CDF/DOC/COMP_UPG/CDFR/7790
version 1.03 September 14, 2005

CDF Run II Annual Computing Plan and Budget for the FY 2006

CDF Collaboration

Abstract

This note presents a yearly update of the CDF Run II Computing Plan. It describes computing strategy, infrastructure and budget requirements for the FY 2006 and projects the needs of the CDF experiment for the next 2 years after that.

Contents

1 Introduction

Run II of the Fermilab Tevatron started in March of 2001. and up until now the collider has delivered over 1 fb^{-1} of proton antiproton interactions. The CDF experiment is recording the most interesting of those interactions, reconstructs the events, and analyses them. In fiscal year 2005 CDF almost doubled the recorded luminosity. Compute systems need to keep up with the ever increasing data and analysis demands. The capacity of the system is increased when needed as to benefit most from technological advances.

We have compared the plan of this fiscal year (as projected last year) with the actual computing upgrades made during the year. We include what we learned from this into the plan of the next three fiscal years. We have updated our computing plan, estimated data storage, processing, and analysis requirements, developed a procurment plan, and estimated the budget to implement it. However, our understanding and projections of the analysis needs are quite incomplete and while we are committed to the long term plan described in this document, the individual projections should be taken with a grain of salt. We will update the CDF Run II computing plan again in a year or before, shall significant changes occur.

1.1 Requirements Model

For the computing planning of fiscal year 2006 we use the requirements model [**?**] developed for the FY-04 planning and later used to plan '2005 computing budget [**?**]. The study uses three models: a"baseline" update of an old model [**?**], and a "single-user" and a "multi-user" model that introduced a new scaling behavior to the requirements. The CDF requirements we will use for our budget and procurement plan come from the "multi-user" model. Updating of the parameters used in the models as well as the models themselves has not been possible this year. While we have good usage and utilization information for the interactive system, usage statistics has only recently been collected for the CAF batch system and the records of the last months are empty due to a software glitch. We will present here some updated tables, figures, and text from the FY-05 study.

CDF is increasing its online event logging capability. The RunIIb upgrade has a significant impact on offline computing requirements. It is designed to allow CDF to avoid deadtime at high luminosities and to maximize the physics program of the Tevatron by writing additional data that will increase the precision of many measurements. One particular measurement driving the upgrade, B_s mixing, is one of the most challenging and important that CDF is expected to make. In FY'06 upgraded CDF data logger will be capable of writing the data at a rate of 60 MB/sec. First step of the upgrade is already completed, the peak data logger bandwidth achieved currently is $\tilde{4}5$ MB/sec

One of the parameters driving requirements to the offline computing system is the size of the raw data event written by the data aquisition system. Figure 1 shows the raw data event size plotted vs instantaneous luminosity for the data taken in spring of 2005. As expected, event size grows linearly with the instantaneous luminosity. At $L_{inst} = 0.5e32$ average event size is $\tilde{1}40$KB, for instantaneous luminosity twice as large ($L_{inst} = 1e32$), average event size becomes $\tilde{1}60$KB.

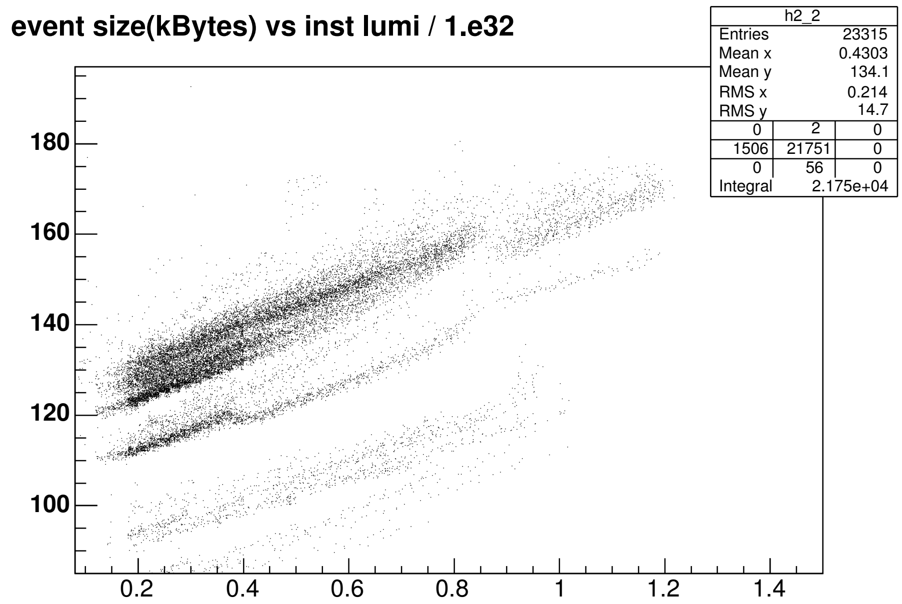

Figure 1: Raw data event size versus instantaneous luminosity for spring'2005 data, (streams B,C,E,G,H,J)

While estimates of the computing resources defined by the needs of the primary offline reconstruction are rather straighforward, it is always difficult to model the requirements to the computing resources coming from the analysis needs. To estimate those we assume that the analysis techniques and data access patterns are stable enough and are not going to change dramatically, and, therefore, needs in the analysis CPU and disk storage resources scale linearly with the total size of the dataset collected by the CDF experiment. We use real parameters of the 2005 CDF computing system to normalize to.

Some of the basic assumptions used in the model calculations are shown in Table 1. Included are the integrated luminosity delivered by the Tevatron, average initial luminosity of a store, the bandwidth of the data logger, average event size, and peak and average data recording rate. The Tevatron luminosity values correspond to the "design" values [1] quoted by the Beams Division.

The experiment rarely operates at the peak event logging rate. More typical values are 60% to 80% of the effective peak rate. We will assume that the average logging rate is about 70% of the effective peak rate.

Bulk of the data is taken at the luminosities significantly lower that the peak ones. Fig. 2 shows distributions for the instantaneous luminosity and the event size for the data taken in spring 2005.

For FY'06 we assume that the average instantaneous luminosity during the data taking is 1e32 and corresponding average raw event size 160 KB.

For each input dataset Production Farm writes the processed events into one or several

Fiscal Year	03	04	05	06	07	08	09
Delivered Luminosity (1/fb)	0.2	0.35	0.6	1.5	1.7	2.0	2.1
Integrated Luminosity (1/fb)	0.33	0.68	1.2	2.7	4.4	6.4	8.5
Initial Luminosity ($10^{31}/cm^2s$)	5.5	6.2	10.5	22.4	27.5	27.5	27.5
Data Logger Bandwidth (MB/s)	20	20	35	60	60	60	60
Average Raw Event Size (kB)	220	150	140	160	160	160	160
Peak Event Rate (Hz)	80	130	230	360	360	360	360
Average Event Rate (Hz)	50	80	170	250	250	250	250

Table 1: Operating parameters and basic assumptions used in the requirements model as function of fiscal year.

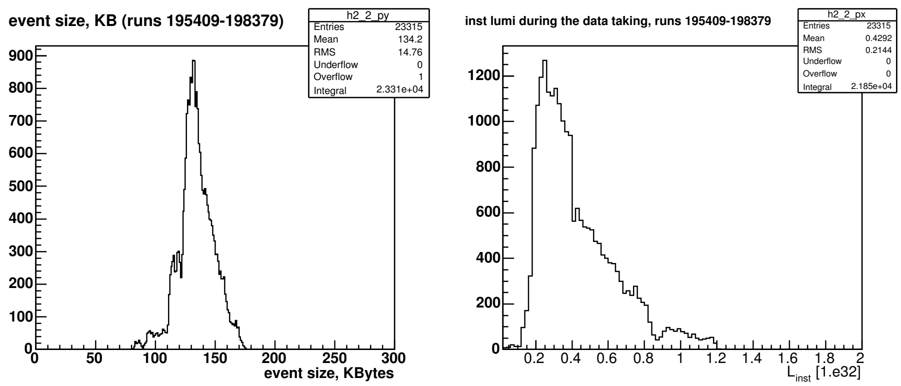

Figure 2: Raw data event size and instantaneous luminosity for spring'2005 data, (streams B,C,E,G,H,J)

output datasets based on the trigger information. Due to the partial overlap between the output datasets and the increase of the output event size the total data volume written by the Production Farm is about 1.4 times larger than the input data volume. It is interesting to note that the ratio of the number of events written to the output datasets to the number of input events is , as shown in Fig. 3 also close to 1.4 - this coincidence is due to the fact that the real event overlap between the output datasets is less than 10%, however when processing 2 out of 6 input data streams (B-physics triggers) the Reconstruction Farm along with the reconstructed data in full DST format outputs so-called “compressed” datasets with the event size about 3 times smaller that that of the full DST event. Events in “compressed” datasets, however, have significant amount of non-tracking information dropped and could not be used for high-Pt analyses.

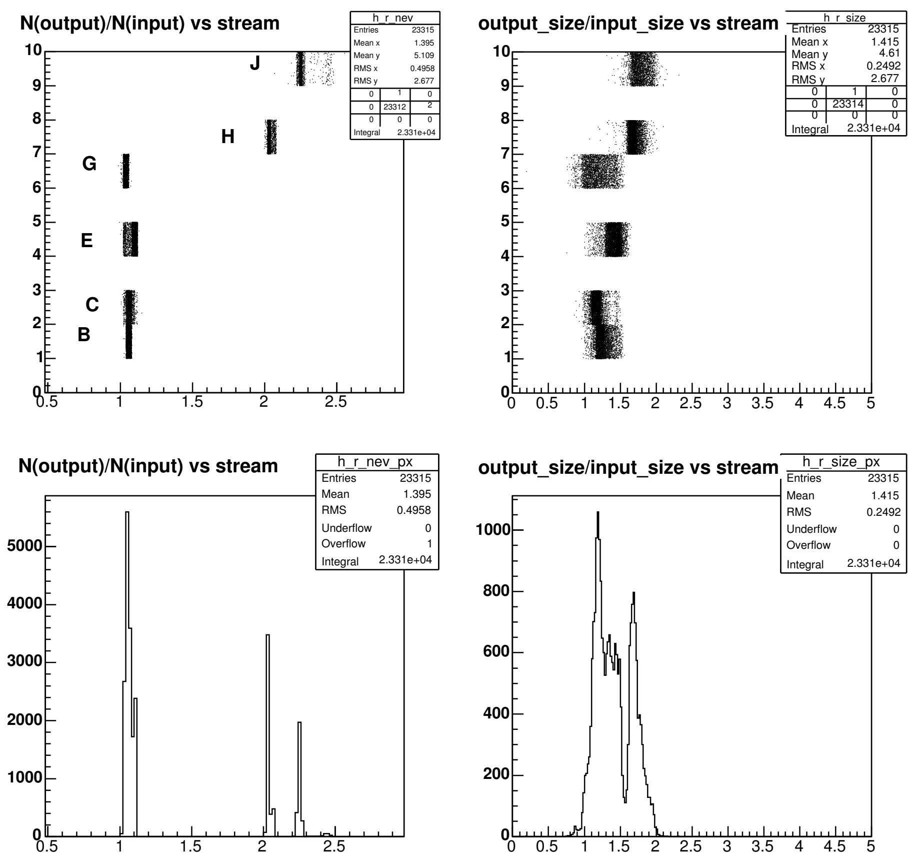

Figure 3: CDF Production Farm: Output/Input ratios for the number of events (left) and the total data volume (right) for different data streams

2 Computing and Analysis Model

A conceptual view of the major computing elements and data-flow at CDF at FNAL is pictured in Figure 4. Although incomplete, Figure 4 presents some of the main themes of CDF computing. Raw data is acquired online and is written to a *write disk cache* before being archived in a *tape robot*. The raw data is read by the *production farms*, either by triggering a cache-to-cache copy or directly from the tape robot, where it is reconstructed and the resulting *reconstructed* data is written back to the tape robot. In both cases, there are caches that decouple the *production farm* from the tape robot. The production farms use calibration constants replicated from the online *database* to the offline database and any other replicas (all shown as one database for simplicity). The reconstructed data is read primarily by *batch CPU* via a *read disk cache*. Some of the reconstructed data, and the majority of secondary datasets from the reconstructed data, are also stored in the disk cache with relatively large cache lifetimes where they are accessible by the batch CPU. The batch CPU produces secondary datasets and root *N-tuples* and writes them to output disk and also the tape robot via other *write disk caches* (distinct from read disk caches). The batch CPU makes extensive use of the offline database and its replicas. The batch CPU also analyzes the N-tuples on the static disk. *Interactive CPU* and *user desktops* are used to debug problems, link jobs, and send them to the batch CPU which is the workhorse of CDF analysis. The user analysis farm is exclusively batch. Users desktops can also obtain data from the tape robot via read disk caches, write them back to the tape robot via write disk caches (not shown), and transfer N-tuples and results back to their desktops from the interactive and batch CPU. User desktops and interactive CPU make use of the offline DB and its replicas.

In this model physics groups are encouraged to utilize the batch CPU to produce secondary datasets and write them to static disk and the tape robot. Users are encouraged to produce N-tuples on the batch CPU and transport them back to the desktop for further analysis, but also have the option of utilizing the batch facilities for subsequent re-analysis of the N-tuples. Users have access from their desktops and the interactive CPUs to the datasets on the CAF output disks. The interactive CPU provides a controlled environment for debugging and job submission. The upgrade of the interactive CPU is discussed in Sec. 3.

Offsite resources contribute to this picture by adding additional CPU and disk caches. However, we do not expect to be using offsite tape archiving facilities at this point. The tape robot at FNAL thus serves the role of central storage facility for all official CDF data. In contrast, we do not require a copy of user level data to be stored centrally at FNAL, nor do we require tape storage prior to general open use of the data in CDF. More details on our future vision of bluring the distinction between offsite and onsite computing are discussed in Section 9.

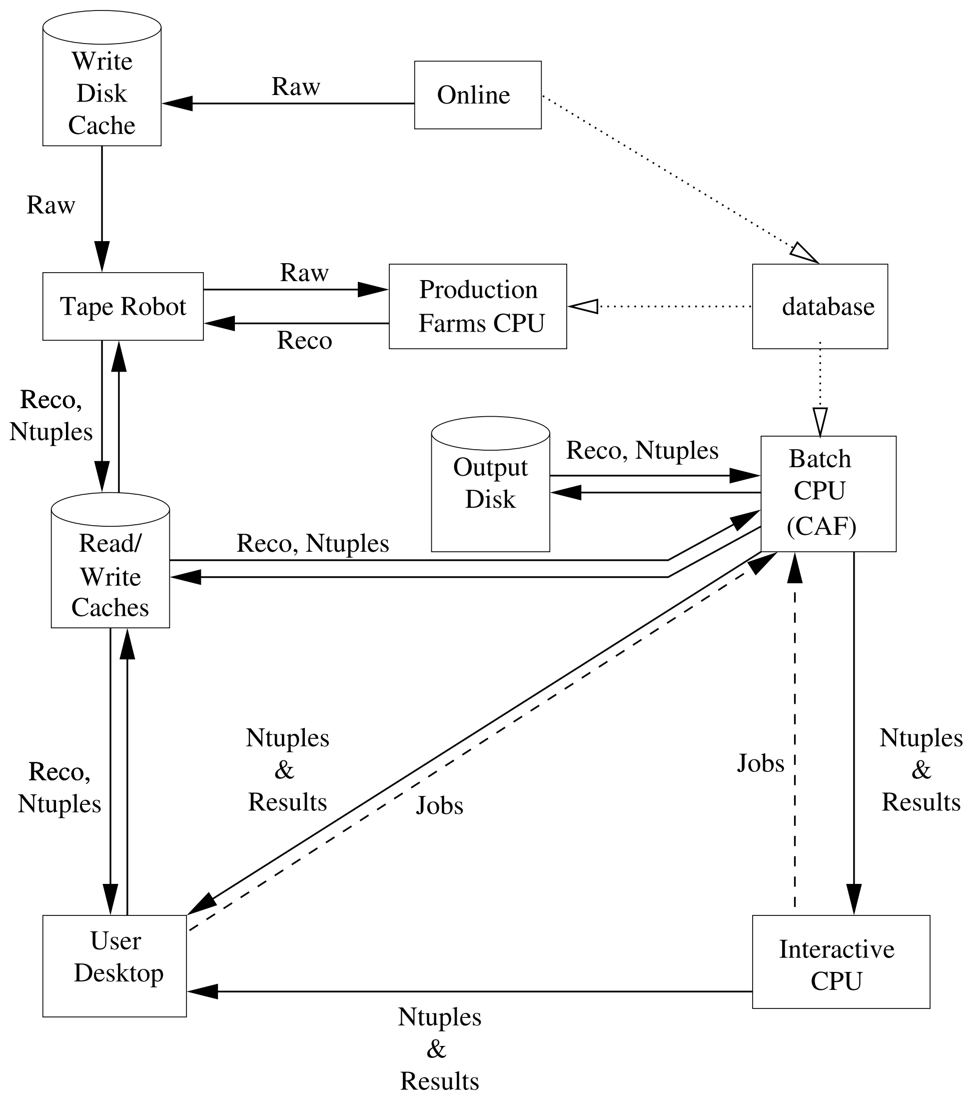

Figure 4: A simplified picture of the CDF Computing Model. Major computing elements in boxes, raw and reconstructed data flow indicated by solid lines, database constant flow indicated by dotted lines, and job flow indicated by dashed lines.

3 Interactive Systems

The CAF discussed in the following section is a batch computing engine that satisfies the majority of CDF's CPU and file serving needs. The CAF is supplemented by an interactive computing system. As of September, 2005, the CDF central interactive computing system consists of three dual-processor 1.4 GHz Athlons (nodes fcdflnx4, fcdflnx6 and fcdflnx7), a dual-processor 2.7 GHz Xeon (fcdflnx5), two 8-processor 700 MHz Intel SMP machines (fcdflnx2 and fcdflnx3), a NetApp machine (fcdfhome) serving 269 GBytes of disk for user home areas and a second NetApp machine (fcdfspool) serving 645 GBytes of disk for user spool. The four dual-processor machines each have 4 GBytes of RAM. An additional 34 TBytes of disk is nfs mounted across the entire interactive pool. All the machines in the interactive pool run Linux. These systems, collectively referred to as the "interactive pool", define the reference environment for offline software. Every CDF user has a common account across the pool nodes, with the exception of the fileservers and fcdflnx3, which is reserved for offline operations.

About 380 Linux/Intel computers in the CDF trailers provide the bulk of interactive computing capacity for the experiment. A small number of special-purpose machines also falls into the category of interactive systems in part because they support some type of interactive capability, and in part because they do not fit into any other category. These systems will be discussed later in this section.

The interactive pool was developed and deployed during the later part of FY2004 and early FY2005 with the goal of reducing operating costs and improving scalability relative to the aging legacy systems based around SGI SMP machines. A commodity computing solution meets both of these goals. The two legacy SGI machines (fcdfsgi2 and cdfsga) were retired over the past year, thereby shedding about \$270k/year in maintenance contracts. With no remaining SGI machines among the offline systems, we have reduced the number of platforms supported and created a more uniform computing environment spanning the central interactive, central farm and trailer computers.

There are three logical components in the interactive pool, as illustrated in Fig. 5: an NIS cluster for account management, the interactive nodes and a bank of file servers. The NIS cluster is constructed using three CAF Stage I dual AMD machines. One machine acts as the NIS Master where accounts are added and modified. NIS maps are created here and dispatched to the two redundant servers. Clients use broadcast requests to spread queries across the available servers.

The interactive nodes, by design, can include any machine that operates within the reference Intel/Linux distribution, Fermi Linux 7.3.2. The four dual-processor machines form the core of the interactive nodes in the pool. A 2 TByte file server provides a common scratch area for all interactive nodes.

A bank of file nfs servers, the third component of the interactive pool, serve several disk arrays that were inherited and migrated from fcdfsgi2. Presently, we have six 2 x 3.2 GHz Xeon machines each with 4 GBytes of RAM and dual gigabit network ports serving the arrays.

The largest interactive computing system at CDF is the collection of desktops in the

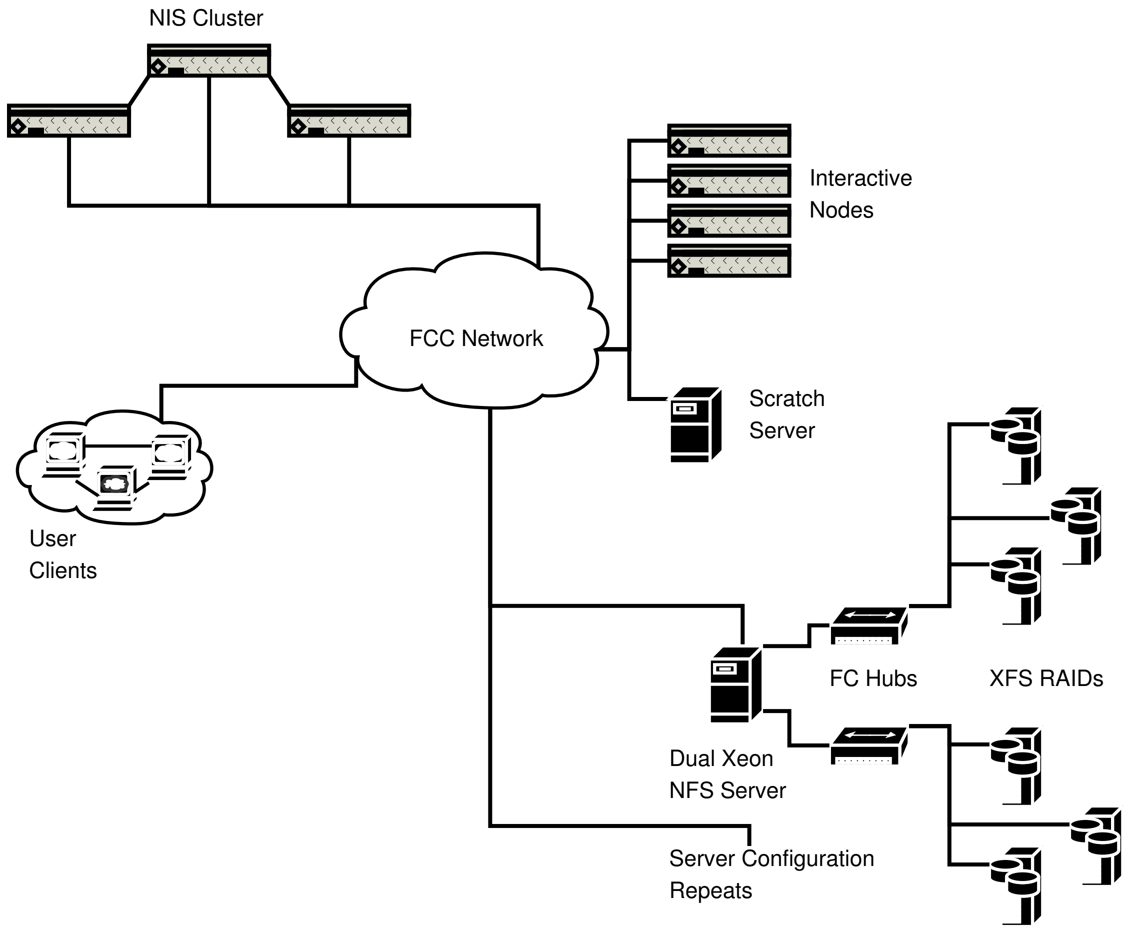

Figure 5: Configuration of the login pool.

CDF Trailers. Currently, 384 Linux/Intel desktops are managed by 1.5 full-time CD system administrators. The availability of this and other off-site interactive resources greatly reduces the demand for central interactive computing facilities, and therefore reduces the required size of the interactive pool. The vast majority of desktops are part of clusters owned by collaborating institutions. Some of these clusters are loose-knit collections of "independent" PCs while others have dedicated file servers (serving home areas and data volumes) and compute nodes. There are also a small number of specially designed clusters managed by institutional and PPD personnel, e.g. the MIT cluster and the ATOM cluster. The growth of such clusters is limited by the available power and cooling infrastructure.

Finally, a number of machines are included within the category of interactive systems, but do not necessarily allow interactive logins to general users. Examples of these machines include the offline web server, code build and distribution machines, code servers, etc. Such machines are considered to be part of the interactive infrastructure of the experiment.

3.1 Plans for interactive systems

Given the large interactive computing capacity in the trailers and off-site, we expect only modest increases in demand on the central interactive systems. At the time of writing, the four dual-processor interactive nodes will be replaced by four new, dual-processor 3.0 GHz Intel machines currently on order. In FY2006, we will purchase another four dual-processor machines (or the equivalent in current technology) to expand the pool at a total cost of about $15k. Similarly, we anticipate procuring up to four nodes per year to expand the pool or replace retiring nodes.

The offline operations machine, fcdflnx3, will be replaced by a dual-processor 2.5 GHz Xeon that was previously used as a head node for the production farm. This type of internal resource transfer demonstrates another key benefit of a more uniform, commodity-based computing infrastructure.

The home area disk space will expand to nearly triple the current size from purchases made at the end of FY2005. The disk arrays originally on fcdfsgi2 are aging and will likely require replacement in the near future. A larger user scratch space hosted on a more reliable platform is also needed. We expect that two fileservers of the class purchased at the end of FY2005 plus two or three head nodes would satisfy both requirements. The cost of this set of systems is about $40k.

CDF is investigating the possibility reconfiguring and organizing a significant fraction of the project disk (i.e., non-dCache disk) into a virtualized disk pool based around the resilient dCache product. Should such a system be deployed, it is likely that the 34 TBytes currently mounted on the interactive pool would migrate into the common virtualized pool. We assume, however, that the disk needs for the interactive pool would remain unchanged under this scenario, and that the above analysis and cost estimates still hold.

Replacements for two of the interactive infrastructure machines will need to be purchased in FY2006. Code servers fcdfcode1 and ncdf209 will be four years old in the early part of calendar 2006, at which time they will be retired. We estimate a total cost of about $15k to replace these machines with the required server-quality hardware. The code build node (cdfcode) was replaced in FY2005 with hardware procured in the middle of calendar 2004. The new system may last for the duration of CDF code development.

To date we have observed no scaling limitations or performance bottlenecks in the interactive systems, associated disk arrays or networks. Many options exist to mitigate such problems should they develop. High fileserver loads can be addressed by deploying more performant shared filesystem technologies, such as gfs, or by sub-dividing disk arrays across more servers. A large demand for data access via rootd could cause an excessive CPU or network load on one or more of the interactive nodes. A set of dedicated rootd servers based upon worker-node type machines would provide a low-cost method to distribute this load. Finally, adding nodes to the interactive pool can be accomplished quickly and at low cost by re-assigning existing CAF nodes. Adding $15k in contingency to cover these and other unanticipated issues, we arrive at a total budget of $85k for interactive computing.

4 CAF Batch System

The work horse for CDF user analysis is a computing cluster presently consisting of $\sim$ 1600 CPU's, adding up to a total of 3.8 THz of CPU cycles, accessing $\sim$300 TB of disk space. These resources as well as the offsite resources [?] are accessed using the CAF software interface to an underlying batch system.

The CAF is a GRID portal for CDF users which unifies many different resources into one simple interface. It manages the parallelization of the jobs including the management of input and output sandboxes, monitoring at system and user level, user interaction and diagnostics, and has provided a model for sharing and allocation of computing resources. The CAF also implements interfaces to data handling, data base, and software distribution services external to the batch system.

We currently have two implementation of the CAF software one for FBSNG and one for Condor. Also in development is an interface to the LCG resource broker [?]. In the past two years, we have converted all of the FBSNG-based systems to Condor-based systems (CondorCAF). Much of this transition has been transparent to the user, who would only notice the improved monitoring web pages of the new system.

In this Section we focus on the services the CAF provides, and briefly describe some outstanding development issues, as well as human resource requirements.

For implementation issues we refer the reviewer to the extensive online documentation at cdfcaf.fnal.gov. While the general design and user-interaction has not changed from the FBSNG-based system described in the original design document and user guide, new installation and operations guides have been written to describe the CondorCAF system. The design of the CondorCAF is discussed in CDF note 7088. In addition, we have found using a Wiki web page to be very useful for maintaining an electronic knowledge base on operational issues regarding site installations, CAF, Condor, dCache, common user problems, and development hardware usage.

4.1 CAF services

The CAF grew out of the need to maximize the amount of computing we can provide for CDF at more or less fixed cost both in terms of hardware as well as human capital to operate the system. Fiscal pressures as well as the scale of the CDF computing challenge lead to a large batch based cluster of commodity PC hardware.

A user compiles, builds, and debugs their application on their desktop anywhere in the world. To do so we provide low bandwidth access to all CDF data files from anywhere in the world interactively. They then submit their job to the a CAF a top-level shell script to run the job, a directory structure that contains all executables and auxillary files, and the level of parallelization desired. The CAF user interface forms a gzipped tar archive and sends it for execution to the specified farm. At the execution site, the user tar archive is submitted to the batch system as many times as was specified by the user at submission time. At execution time, the archive is unpacked, and the user's shell script is invoked with whatever input parameters declared at submission time. One of the input parameters is an integer to distinguish between different instances of the same archive. It is then up to the user to implement the details of the parallelization based on this integer.

After the user shell script terminates the CAF creates a tar archive of the user working directory on the local node in the cluster, and copies it to a location defined by the user at submission time. In principle, the output location may be anywhere in the world. In practice we provide 50 GB scratch space per user inside the CAF. This scratch space may be accessed transparently using a set of environment variables defined by the CAF for the user. The user may access their scratch space via ftp and rootd from outside the CAF, and via ftp, rsh, rcp, fcp, and rootd from inside the CAF. We refer to this as *icaf* to indicate that the intended use is as staging area for CAF output, much like imap for email.

The CAF is thus receiving one tar archive with the application, and sending out as many tar archives as there are instances of the user application requested at submission time. An intelligent user will thus copy or delete all files from their working directory before exiting their shell script except for log and core files that they want back.

While the CAF is fundamentally a batch based system, we were unwilling to sacrifice the core functionality provided by an interactive system. We thus implemented not only the usual batch functionality of *submit, stat, kill*, but also a core set of services that allow a user to watch jobs as if they were running on a local desktop instead of a remote cluster. Among these services are *ps,ls, tail, top,* and *debug*. The first three allow the user to obtain information about the local environment in which a given instance of a job is executing without the need to know where that environment is located. The user need only specify the instance and submission ID to get this information. The debug service allows the user to attach a gdb session to a running executable. To do this, the user needs to specify the Unix PID in addition to section and job id. The user may look up the latter on the CAF monitoring pages or with the interactive *ps* command.

As part of the CondorCAF, new detailed monitoring web pages have been developed which allow user and administrators to understand the current state of the system. This includes monitoring of the CPU and memory consumption of jobs, the user priorities on the batch system, the break down of jobs according to their data handling usage and many other useful diagnostics. Among the other details, the web-based monitoring provides the CPU time consumed for each process spawned by each instance of a user's job while it is running.

Once all instances of a given submission have terminated, the CAF will parse a set of CAF logfiles created for this submission, and write a summary report to be emailed to the user. The objective with this email report is to provide the user with a quick overview of how well their submission completed. The body of the report provides sufficient information for the user to determine which instances have failed, as well as the reason for failure if known. It is thus very easy for a user to go back and debug individual instances by either inspecting the core and log files they received back with the output tar archive, or by running a specific instance interactively through a debugger. A detailed I/0 monitoring report is generated for each job detailing which files were read/written and the corresponding amounts of data.

These diagnostics are also archived indefinitely, which has proved useful in determining future demands and creating load estimates for the data handling system.

In the past year a few new user specified parameters were added. First the user can specify an accounting group. This corresponds to a virtual set of resources on which the user has special priority. These priorities are given either based on institutional purchases or to reflect CDF priorities (e.g. high priority data validation). A CAF interface to the data handling system has also been added, which checks that the users data set exists and whether it will be staged from tape or is already on disk. The users it warned if they will be consuming a large ammount of tape drive resources. The dataset parameter is also used to configure SAM projects as describe in the external services connection below.

We consider the CAF interfaces to be in their final form except for minor modifications. However implementation of these interfaces has and will continue to evolve with CDF GRID deployment [**?**].

4.2 CAF interfaces to external services

In the past year the interfaces between the CAF software and external services has continued to develope. In particular, the data hanlding interface for SAM and the database interface for the Frontier system are now included. These interfaces allow the manager foreach DCAF site to specify in the configuration the locations of the local Frontier and SAM systems.

For the SAM interface, the CAF software manages the execution time of the start of SAM projects so that they are started at a time near when the computing resources will be available. When a jobs is completed, the SAM system is notified and a summary of the status of the project is reported to the user in the email.

The SAM interface also allows the user to specify different predetermined SAM configurations which can be used for testing or to allow users to access special SAM stations which are configured to access different data handling resources.

Further development of this interface may be useful as we gain greater experience with using SAM on the CAF.

4.3 CAF future directions

We believe that the CAF's long term value lies in its services provided to the user, as well as its monitoring. The lasting intellectual value is thus in concept rather than implementation. Implementation while its cardinal weakness is also a crucial strength. It is entirely home brew with no standards other than kerberos used in its implementation. This allowed us to build the first system in little more than 6 months. We are now in the process of developing new backends to the same simple interface, which allow us to exploit a wide variety of GRID resources, while maintaining a simple and useful user experience.

The computing requirements of the CAF increase with increasing luminosity and trigger rate as described by the CDF computing model. Table 2 shows the expected CAF resources as a function of time.

As infrastructure and budget limitations prevent all the CAF equipment from being housed at Fermilab, CDF has begun implementing a distributed computing model. In 2004 about 25% of the CPU resources were located off-site, as it is seen from Table 12 by the summer of 2005 fraction of offsite CPU resorces increased up to about 50%.

However with the approaching LHC startup amount of the CDF-dedicated resources outside Fermilab is expected to decline and CDF will have to rely on using those resources in the shared mode.

For today's planning purposes we assume that due to the early deployment of the GRID interfaces CDF will be able to get enough CPU resources to generate all the necessary for analysis Monte Carlo datasets offsite.

To estimate CPU resources needed for MC generation we assume that it takes 30 GHz*sec to generate a typical MC event (this estimate comes from the Pyhia dijet MC) and use the fact that in 2004 CDF has generated about 400 Mln MC events. We also assume that the required MC statistics has to be generated within the 2 months.

Predicting the CPU trends we replace Moore's scaling (x2 in 18 months) which is over-optimistic and is not followed to by the assumption that the CPU clock speed increases by a factor of about 1.3 which is in good agreement with the evolution of the CPU clock speed over the last 4 years.

Fiscal Year	Total Need (THz)	MC CPU needs (THz)	On-site CPU (THz)	Reco need (THz)	Off-site CPU (THz)	New On-Site (# CPU)	Retire On-Site (# CPU)	CPU Speed (GHz)	Cost On-Site ($M)
03	1.5	0.6	2.2		-	2* 159	0	2.2	0.31
04	2.7	1.3	2.6		-	2* 200	2* 31	2.8	0.49
05	7.3	3.5	4.4	0.5	-	2* 300	2* 200	3.6	0.66
06	9.9	4.6	6.5	0.5	-	2* 300	2* 256	4.6	0.66
07	18.4	7.9	8.4	2.2	2.1	2* 300	2* 242	5.9	0.66
08	31.5	12.3	10.6	6.5	8.6	2* 400	2* 244	7.5	0.88

Table 2: CAF annual requirements for on-site and off-site resources. "on-site CPU" : projection for the CPU available at FNAL "off-site CPU": projection for the needs in the off-site analysis CPU

Table 2 shows that for through the all of Run II CDF can run offline reconstruction using Fermilab resources only, however from 2007 significant fraction of the CDF analysis jobs will have to run outside the Fermilab. This requires development of the GRID technologies allowing off-site analysis of the large datasets.

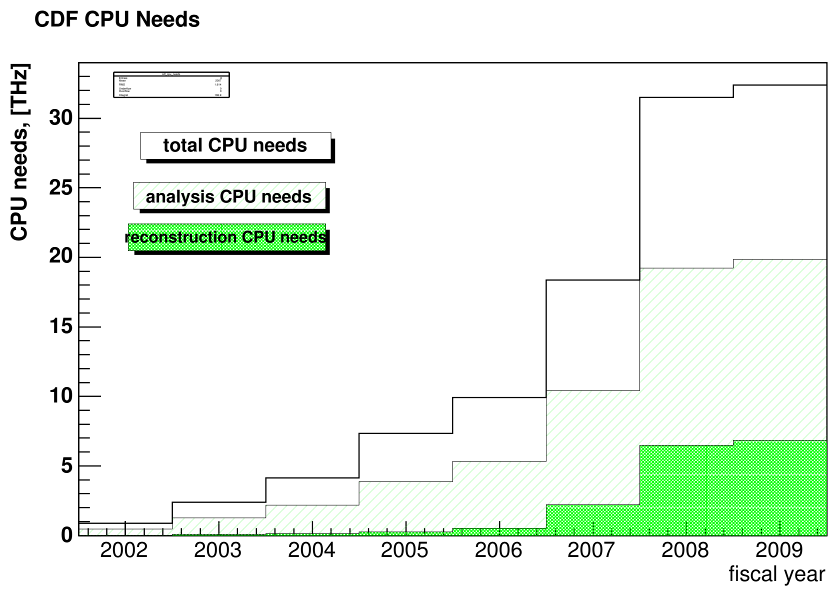

Figure 6: CDF needs in reconstruction, analysis and MC CPU

Figure 6 shows CDF projections of the CDF needs in CPU cycles for different kinds of jobs

5 Data Handling and SAM

The CDF DH (Data Handling) system is comprised of user application interfaces (DH modules in AC++), SAM, dCache, and Enstore. One may think of these four elements as user API, "data handling", cache management, and archival storage. SAM's role in the CDF DH system is to control data movement, and to record this movement in the meta-data catalog.

Support for the elements of the DH system is divided among several entities: The DH modules are the responsibility of CDF, and are currently supported by the CDF project within the Run II department in Fermilab-CD. SAM is a joint CDF and D0 project recently joined by Minos supported predominantly by CD-Run II Data Handling group, with database and GRID support from CD-CSS. On the CDF side, Data handling efforts were/are supported recently by the Duke University and UK and Italian UK collaborators. Routine operation of CDF dCache is the responsibility of CD-Run II, with development support from CD-CCF. Operation of CDF Enstore system is the responsibility of the CCF department.

The last year had continued to see quite significant changes in the DH system, with the focus being the deployment of SAM for production use of and gradual retirement of the replaced elements of the DH system as well as v5 to v6 (and subsequently v7) SAM migration.

The remainder of this section is organized as follows: We first discuss archive related costs, as well as the model used to predict them. Costs for cache disks are discussed in Section 5.2. This is followed by a discussion of DH operations and performance. We conclude with current state of the SAM deployment and future directions.

5.1 Data Archive

The tape archive consists of three components: the automated tape library, the tape drives that provide I/O to the archive and the tapes that fill it. In this section, we will discuss the requirements relevant for each of these components and discuss the plans for meeting those needs.

5.1.1 Data Archive Requirements

The tape archive must accommodate the raw data from the detector, the primary production datasets, secondary datasets and Monte Carlo data, all of which are EDM-based root files. This accounting neglects the volume contributions from tertiary datasets or other highly compressed files created by the physics groups, although their volume and especially bookkeeping starts to be recognized as an non negligible component of the data.

We will first provide the numbers for the FY-05 and then make an extrapolation into the following years.

The volume of the raw data as of September 2005 collected since December 2004 was about 113 TB. It corresponded to about 621 9940B tape volumes. The ratio of the corresponding primary production datasets to the raw datasets is about 1.15 in terms of data and about 1.25 in terms of tapes.

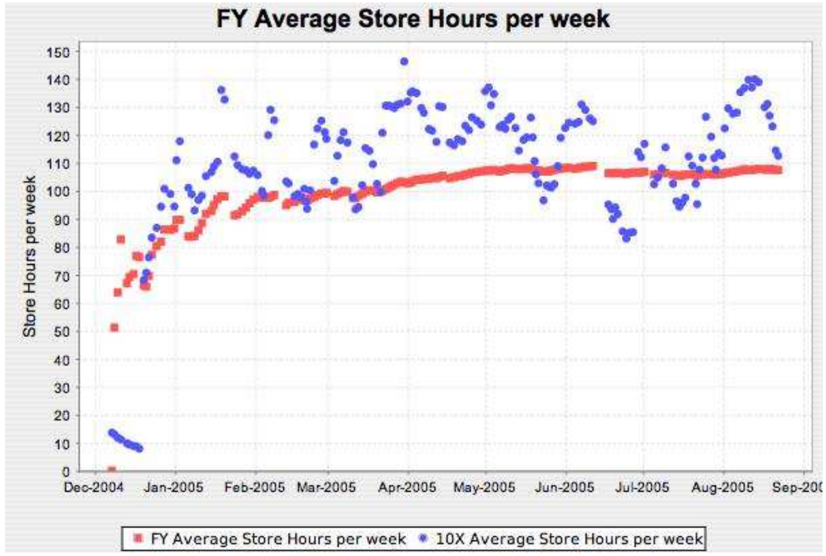

Figure 7: FY-05 CDF Average Store Hours per week.

The number of the store hours averaged at about 120 since February (Fig. 7). The detector up time was predominantly above 90%. The number of store hours per week was 120h/per week recently. At 95% efficiency gives it 68% data taking duty cycle. The actual efficiecy was about 81% on the average. Given the above the average data taking rate times the duty cycle was aout 5MB/s averaged over the entire FY-05 starting December till mid September 2005.

The volume of the raw data per week as the function of the store hours times CDF detector up time can approximated by: $3 \cdot 10^{10} * \text{liveTime} + 10^{11}$bytes for the recent stores. This yields about 20 9940B tape volumes per week. This would result in about 124TB and 680 raw data tapes in the entire FY-05.

The average level-3 data rate rate is a function of the peak data logging rate from the experiment. Upgrades to the peak logging rate proposed in FY-03 are being implemented now almost doubling the rate from $20\,\text{MB/s}$ to $45\,\text{MB/s}$. The rate is planned to increase again in FY-06 to $60\,\text{MB/s}$, where it remains for the rest of Run II.

The total I/O demands on the robot will determine the number and type of tape drives that are required. To estimate the I/O to the archive, we sum the contributions from all sources: writes of raw data, farms output, re-processing, secondary datasets and Monte Carlo storage, and reads for production, secondary dataset creation and general analysis. The contribution from the completed tape migration from 9940A to 9940B tapes was small. The load due to future media conversions is not specifically included.

Data moving in or out of the archive is staged to disk first in order to adapt the I/O rate of external data consumers or producers to the I/O rate of the tape drives. This staging step implies that the archive need only provide the average read and write rates in order to keep pace with demand. To obtain the bandwidth required by raw data logging, for instance, we

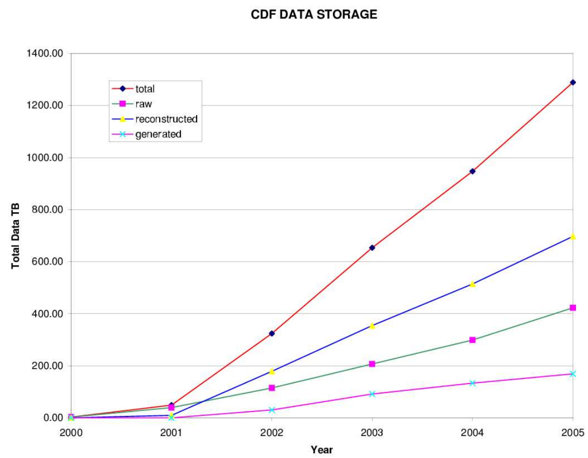

Figure 8: Volume of raw, reconstructed and simulated data stored in the tape robot as a function of time (calendar years). The total volume is shown in red. 2005 point is extrapolated to end of September

multiply the peak logging rate by the operating efficiency during peak periods (typically 0.7). The data rate required to write output from the production farm into the archive is obtained by multiplying the raw data write rate by the ratio of production output to raw data event sizes.

At present, raw data processed on the production farm is written to the archive, then read back to the farm directly from tape, requiring corresponding tape drive capacity.

To estimate the archive I/O required by user analysis, we take the total estimated read rate on the CAF and multiply by the cache miss rate. Experience indicates that about 10% of the file requests on the CAF result in cache misses that require reads from tape (see Sect. 5.3).

The results of these estimates as a function of fiscal year are presented in Table 3. In FY-05 raw data and farms output account for write rates of about 2 TB/day. Rates are sustained at around 4.5 Tb/day in late 2005 while processing all FY-05 data. Read rates peaked at around 16 TB/day, limited by tape drive availability.

To determine the number of tapes needed to provide the required archive capacity, we consider not only the size of existing tapes, but also anticipated changes in tape technology and available densities. Such developments occur over long time scales and require careful planning of technology evaluation, deployment and possibly density migrations.

A migration of CDF data from the old 60 GB 9940A density to the 200 GB 9940B density was completed in FY-04. The process was performed over 18 months at low priority in order to avoid interfering with normal tape operations, and in order to avoid the purchase

Fiscal year	05	06	07
Peak logging rate (MB/s)	20-45	60	60
Raw data (TB)	124	250	250
Production output (TB)	143	285	285
Secondary datasets (TB)	35	190	190
Annual archive (TB/year)	302	725	725
Total archive (TB)	1290	2015	2725

Table 3: Tape archive volume. Data volume is scalead by the peak logging rate. Secondary datsets are sassumed to be 2/3 of the production output in FY-06,07.

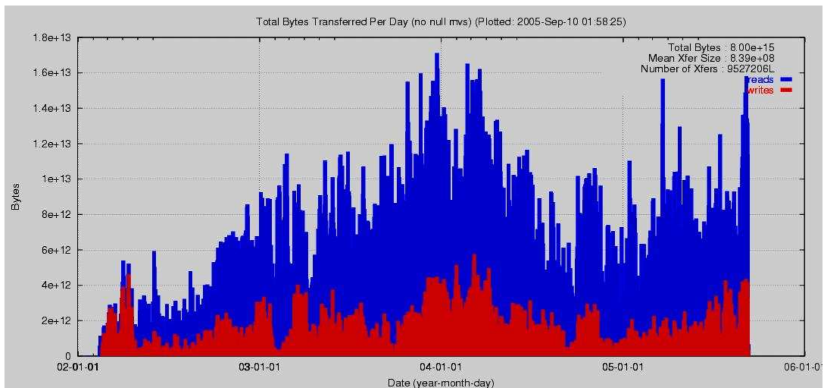

Figure 9: Tape I/O (TB/day) as a function of time. Much of the load in Summer of 2005 was due to the (re)processing of CDF FY-05 data.

of additional expensive tape drives. The process re-cycled about 6000 existing tapes and avoided the purchase of about 4000 tapes over the past two years, which would have been an expense of roughly $300k.

In the 2003 plan, we expected to migrate to an as yet unspecified technology "X" in FY-05 with twice the density of the existing 9940B tapes. This new technology would require the purchase of new tapes, so tape re-cycling will not be an option. To date, these tapes are not yet available which poses a significant challenge.

To calculate the number of tapes needed, we take the estimated archive volume each fiscal year and divide by the tape cartridge capacity. The requirements are shown in Table 4. For FY-04, we estimated a tape consumption rate of about 40 tapes per week averaged over the entire year. Figure 10 shows the volume of data written during 2005 ending in September. The tape consumption rate during the last weeks of the plot is about 120 per week as a result of the farm (re)processing the raw data collected from December 2004. We expect that the total 9420 tape volumes will be used by the end of FY-05 (unless more tapes are recycled or partially filled tapes are appended to).

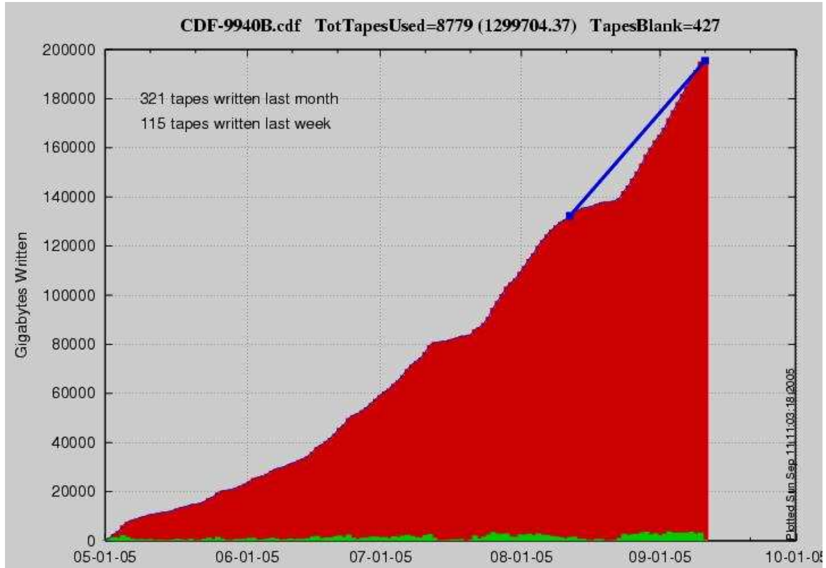

Figure 10: Recent tape usage rates by CDF. The recent tape usage increase is a result of the farm catching up with the raw data reconstruction.

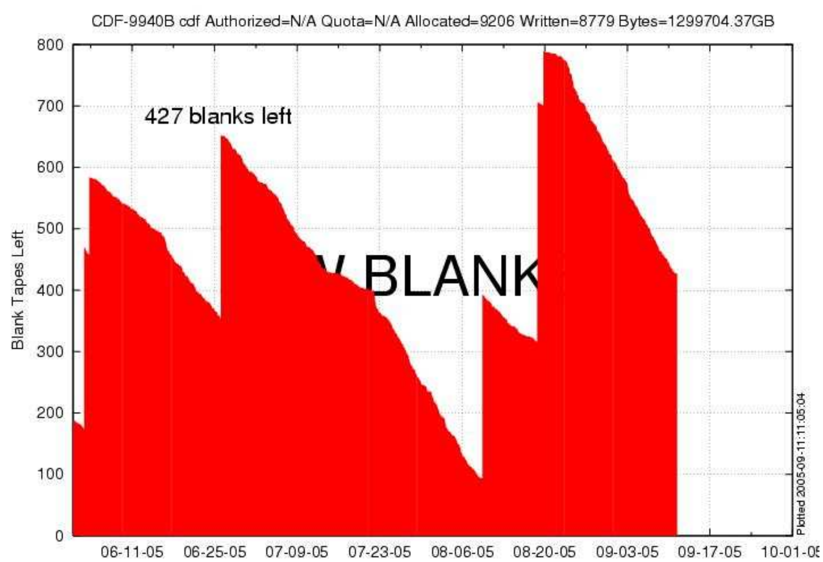

Figure 11: Recent tape usage rates by CDF. The recent tape usage increase is a result of the farm catching up with the raw data reconstruction. The increases of the number of blank tapes are results of tape recycling and purchasing of new tapes.

Fiscal year	05	06	07
Capacity added (TB)	302	725	725
Tape capacity (GB)	193	193	386
Cartridges used (act/est)	2951	3755	1880
Cartridges added/recycled	2332	4130	7330
Migration needs	0	0	5570

Table 4: Media requirements. The recent experience shows that the effective tape capacity is not 200GB but about 193GB including various loses. As of mid September 2005 there were 9420 tapes including 412 blank ones and 1688 empty slots in the CDF STK library. We estimate to be able to add about 4130 tapes by reusing them or by filling up the existing slots. If no new library is added we expect a deficit of about 250 slots by the end of FY-06

5.1.2 Data Archive Procurement Plan

To calculate the number of tape drives needed to operate the experiment, we take the estimated I/O bandwidth to the archive and divide by the I/O capacity of the drives. We then multiply the result by a contingency factor of two to take into account tape drive contention, separation of reads and write, down-times, etc. We ignore any constraints on the total number of drives that can be used by the robots and issues such as the mixing of drives types within a single robot.

Table 5 shows actual and projected drive procurements through FY-07. The current archive uses STK T9940B drives, with a maximum I/O rates of 30 MB/sec (25MB/sec effective rate).

FY	Needs (MB/s)	Robots Total	Drives Bought	Drives Total	Date Avail	Storage (PB)	Rate (MB/s)	Cost ($M)
03	190	2	3B	13B	1/04	0.64	400	0.20
04	410	2	5B	18B	7/04	1.0	540	0.13
05	940	2	13B	31B	7/05	2.0	930	0.43
06	1900	2	16X	31B+16X	7/06	3.0	1900	0.48
07	3000	2	19X	31B+35X	7/07	4.0	3000	0.57

Table 5: Robot procurement plan. Numbers in FY-03 and FY-04 are actual and FY-05 to FY-07 are estimates. As of September 2005 the number of drives is still 18

During FY-06, we will seek ways to reduce the number of tape drives needed by the experiment. The motivation for this effort is two-fold. First, the cost of drives is a large fraction of the total computing budget. Reducing the number required potentially frees funds for other uses or allows us to meet our budget guidance. Second, we are reluctant to spend substantial sums on a tape technology that is about to be replaced with a much more effective technology. The specific steps we will pursue are:

- Actively pre-stage the data for the Farm Input thereby using the tape drives more efficiently. We also expect that once the farm will be following closer the data taken the automated pre-staging of just written data should have a larger effect that it has now where only the calibrators can benefit from it at the moment.

- Expand and reorganize the dCache read pools in order to reduce the cache miss rate, and therefore on the need to read tapes. We also need to replace the aging file servers which start to fail and replace the file servers which were moved to the Production Farm before the new ones could have been purchased.

- Use dCache write pools to decouple data sources from Enstore, and allow optimization of data transfer into Enstore. (This will be needed in any case for next generation 60 MB/sec drives.)

Unfortunately none of the previously expected large capacity tape drives are available yet: While some of these could be installed in the existing STK robots, it will take about nine months to evaluate and certify any of them once they become available. It therefore seems unlikely that we could actually deploy a new technology substantially earlier than very late in FY-06.

To obtain the cost of tapes needed, we first multiply the expected archive volume and number of tapes from Table 4 by a contingency factor of 1.2. Assuming a cost of $75 per cartridge for both current and technology "X", we obtain the actual and projected media costs shown in Table 6. The cost of density migration has not been included.

FY	Archive Volume (PB)	9940A Tapes (PB)	9940B Tapes (PB)	"X" Tapes (PB)	Tape Cost ($M)
03	0.40	.22	.24	-	0.18
04	0.98	-	-	-	0.00
05	1.3	-	.22	-	0.10
06	2.0	-	.33	-	0.13
07	2.7	-	-	2.7	0.51

Table 6: Tape procurements. The fiscal year, data written to 9940A tapes, 9940B tapes, X tapes and the total cost that FY for tape purchases. Numbers in FY-03 to FY-05 are actual and FY-06 to FY-07 are estimates.

Presently we have written to roughly 8,800 of the 9200 tapes in the CDFEN silos, with about 1680 slots available to be filled with new tapes. The projected data logging in FY-05 will fill most of the available 10,900 tape slots in the two existing STK Powderhorn 9310 silos.

Even if higher density tapes become available in FY-06, it is unlikely that density migration can prevent the need for an additional robot.

To deal with continued demand for archive space, we are considering or have implemented the following range of possible actions, listed in approximately the order of preference:

- Remove or re-cycle old data tapes which no longer have physics value. We have started the process of recycling tapes with old and not needed any more production and Monte Carlo Datasets.
- Consolidate or fill little filled tapes (due to an early decision to have one file family even per small dataset) to increase the average tape utilization. We may be able to recover more than 20% of used tapes by using the space on the partially filled tapes. (We had already implemented a more efficient tape file family schema for the newly written MC tapes via DFC mechanism , effectively merging all physics groups tapes into that group file family).
- Remove a large fraction of raw data tapes either to cold storage or to alternate robotic capacity at Fermilab. Once the reconstructed datasets are available for a given file, CDF should normally have no further need of raw data tapes, except for rare technical reasons. This could free up nearly a third of the slots.
- Expand into existing libraries. This requires service agreements, and potential support for an additional robot technology with which we had operational difficulties in the past.
- Purchase another library.

5.2 Network Attached Disk

The basic plan for disk is to store as much processed data on disk as possible while also providing sufficient space for staging, data caching, data validation, and Monte Carlo data storage. In addition to these uses, some disk is required to store N-tuples or other analysis data samples coordinated by the physics groups.

During FY-04, about 150 TB of dCache pools were deployed. About another 50 TB of disk were used for CAF staging or dedicated to local storage for specific university groups.

The majority of the analysis resources have gone to the large B physics datasets. For planning purposes we can scale disk requirements directly with data logging rates. Table **??** shows the estimated disk space needs and cost for FY-05 and beyond, and the actual volume and cost in FY-03 and FY-04.

We do not currently have a good model of the relationship between the total disk space in dCache and the cache miss rate. In the coming months, we hope to improve this understanding in order to better optimize the balance between the amount of disk space and the tape and tape I/O requirements. As previously discussed, reductions in the production event size may change this balance and reduce the need to scale data handling services to still higher levels.

FY	Need (TB)	New Server (#)	Server Size (TB)	Additional Space (TB)	Total Space (TB)	Total Cost ($M)
04	320	8	8	64	340	0.14
05	490	26	14	364	710	0.44
06	710	23	20	460	1170	0.39
07	1170	30	35	1050	2230	0.51
08	2230	19	56	1060	3290	0.32

Table 7: Disk procurement plan at Fermilab. Numbers in FY-04,05 are actual and FY-06 to FY-08 are estimated needs.We estimate that the actual total disk size will be about 10% lower due to the retirement of the old servers

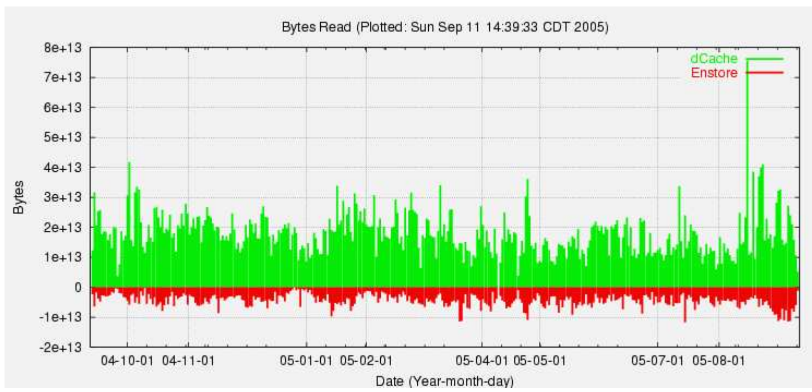

Figure 12: Number of bytes read per day from dCache. Data starts on September 11th, 2004 and spans one year.

5.3 Data Handling Operations and Performance

The dCache and Enstore systems typically handle an I/O load of about 20 TB to 40 TB per day, as shown in Fig. 12. The fraction of data read from tape, shown is red, is usually about 10% of the total data volume delivered unless new data is being reconstructed and then read. Based on special load tests and experience with real user loads, we estimate the existing system can provide acceptable file delivery service at about 80 TB/day, and 4 TB/hour. We have already seen sustained loads of 70 TB per day.

To maximize cache hits, thus minimizing DH related inefficiencies on the CAF, we partition the dCache system into several pool groups, based on the expected access patterns. The usage load in each of these groups has minimal impact on the other groups.

1. "Volatile": regular cache, any datasets not mentioned below.

2. "Golden": secondary datasets that are most relevant for a conference season. We

guarantee that those are always on disk by providing sufficient disk space to keep up with new data coming in. We arrive on the list of golden datasets in collaboration with the CDF physics group conveners.

3. “Raw Data and Big Buffer ”: some datasets, especially raw data streams, are either so large or so infrequently used that the number of times a file is accessed while in cache is rather small. This cache pool thus functions more like a FIFO buffer than an actual cache.

4. “Little Buffer ”: some deprecated datasets should be accessible on a limited basis, allowing only a few files to be accessed and with very limited disk space allocated. We have set three pools for a total of about 2 TB for this group, and tightly restricted the number of tape drives available.

We intend to reorganize the above dCache pool layout effectively moving the disks from the static “Golden” pool to “Big Buffer” and to make the “Golden” pool less static to delegate more of the coordination of the “analysis coherency” to the physics groups.

The data handling system issues a warning to users who attempt to access large datasets that are not yet on disk. We require such activities to be coordinated with the DH operations group so that the data can be pre-staged, thereby minimizing loss of CPU time on the CAF due to tape latencies. SAM users can get some level of automatic pre-staging because a user declares their dataset at CAF submission time rather than at runtime. Eventually, SAM should automate all pre-staging activity.

5.4 Current Status of SAM Deployment

There were many reasons for CDF to adopt SAM:

- Combined development and maintenance of DH software with D0, reducing costs by eliminating redundant solutions.
- A good path to GRID supported tools.
- DH support for off-site computing. This is discussed in detail in section 9.
- Improved operational efficiency of the CAF at FNAL, as discussed above.
- Flexible creation of derived datasets. SAM dataset definitions are created directly by users and groups, the traditional CDF datasets are (mostly) just Enstore file families tracked via the database and file naming conventions. File families are useful administrative tools crucial to efficient tape utilization, but not nearly fine grained enough to track the full range of physics analysis activities.
- Standard, automatic tools to track the processing of files so that partially completed projects can be recovered in spite of occasional hardware and software failures. This is particularly valuable for Farms production and when producing large secondary datasets.

Since the time of the 2004 review CDF had deployed SAM for either full production or a limited use in several areas. In addition a SAM users committee was formed with at least one member from each physics group. The members of the committee had edited, together with the Data handling group a CDF specific users documentation which is the primary SAM document for CDF users. Here is a list of new SAM developments items since 2004 review:

- a lot of effort went into testing and deploying version 6 and version 7 dbservers as well as new station versions and addressing issues of specific usage patterns and isolating problems seen.
- in order to achieve the above a fully dedicated testing facility was established.
- an adequate number of production nodes were made available by reassigning the nodes from CAF which allowed among others to retire an old SUN OS based SAM web server.
- The so called "frozen" python client (v7) was introduce to avoid delays related to loading many python libraries over the network.
- The CDF Reconstruction Farm is using SAM to access the raw data and is storing and declaring all it production output to SAM only.
- SAM is fully integrated for CAF use for read access of all datasets.
- SAM was fully integrated and debugged for use within the CDF analysis framework (AC++) directly with the GCC compiler without relying on the python command line interface.
- the raw data is being declared directly to SAM. However, the system runs on an old SGI machines and needs to be made more robust and therefore the data is also declared to the Data File Catalog (DFC) and a process monitors the consistency of the DFC and SAM raw data information.
- Monte Carlo files generated at Rutgers University are being uploaded directly into SAM although other groups still upload the files via a DFC based mechanism. The data is subsequently entered into SAM. It is mainly due to reasons related to the change in mapping of the Monte Carlo datasets to the tape file families which had not yet been implemented for the SAM mechanism and therefore can not be opened for the general use.
- many datasets are available on remote stations for use on their corresponding DCAF's. The total amount of data at remote stations is about 70TB. The amount of data analyzed with SAM is shown in Fig. 13. The data does include load tests but those were only performed at the cdf-sam and cdf-caf stations and not at the remote ones which show a steady use.

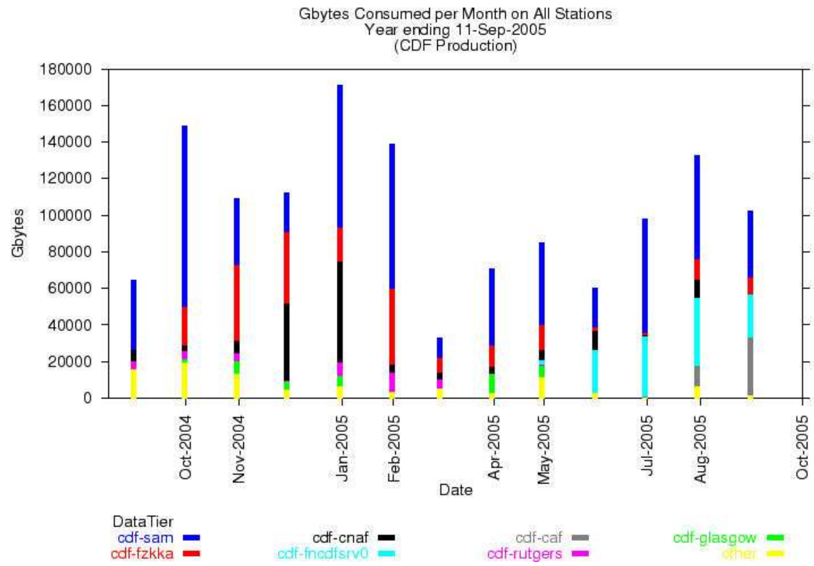

Figure 13: Number of Gigabytes of data read at CDF Stations over the course of the past year ending September 11th, 2005.

- SAM is being used to "skim" large datasets and upload the results via SAM on a limited basis by the Italian collaborators.
- all SAM version 5 stations had been replaced by the version 6 ones.

We expect the following progress to be made in the CDF SAM deployment with the next few weeks:

- the so called CDF "SAM auto-destination" server is going to be modified in order to implement a more efficient file family schema for the Monte Carlo datasets and to use native SAM auto-destination which may need to be modified as well to accommodate the CDF specific needs.
- the above should enable the storage of all the Monte-Carlo production via SAM.
- we plan on shutting down all version 5 dbservers except one used for JIM (Job Information and Monitoring) testing on September 15th.
- we will be upgrading all remaining (and not dedicated to the Production Farm dbservers) to version 7 on September 15th.

We expect an improvement to the SAM dCache interface which should result in a better pre-staging mechanism to be addressed in the next few months. We plan to migrate the farm to the version 7 of the dbservers (using the new test facility in order not to jeopardize the farm reconstruction process) in the next few month as well.

5.5 Future Directions

5.5.1 Durable Cache

Apart from write caching in front of the robot, we also have a clear need for better support of non-archival, but durable storage for individual user data. In a typical analysis, a user starts with some secondary dataset produced in a coordinated fashion by a physics group. The output of this processing on the CAF will generally be a quite sizable collection of relatively small output files. The user thus needs to store these files temporarily for validation, further analysis and possibly concatenation. In general, this processing step is done more than once in order to fix some oversight or the other. Old versions may be deleted to conserve disk space.

An ideal storage system for this use case is disk resident only, and supports deletion as well as reservations and quotas. At present, we support this activity by providing user scratch space inside the CAF. This solution, however, does not scale well, especially if groups of users organize themselves to produce common datasets.

There is an effort to implement these ideas based on dCache software which had recently entered a pilot/testing phase with University of Michigan group being one of the main active proponents of the system.

6 CDF Production Farm

The CDF Production Farm provides the experiment with the reconstructed data. It uses about 300 dual Pentium nodes with the current CPU capacity of about 750 GHz.

Production Farm has several operational cycles:

- beamline production: reconstruction executable which includes track reconstruction algorithms only runs on a primary dataset (so-called Stream G) which includes jet triggers and as such is unbiased with respect to tracking. The output is used to fit the run-dependent beamlines. Size of Stream G is about 15% of the total, the latency of this cycle is 3-4 days.
- standard offline reconstruction executable with the final calibrations runs on all the data. This cycle provides experiment with the reconstructed data to satisfy demands of the CDF physics analyses. A design latency of 4-6 weeks with respect to the data taking has recently been achieved.

In 2005 CDF transitioned from the Production Farm described in **??** to a new architecture. The goal of the transition was to standardize the book-keeping and job submission procedures and thus provide for the scalability of the Farm.

6.1 SAM-Based Production Farm

The CDF Production Farm is using data handling and file metadata services provided by SAM.

Resource management is implemented via the regular cron jobs probing utilization of resource and services.

Job submission is a periodic cron job checking on the resource status and is prohibited if any of the dependence is not sufficient.

Input for the Production Farm jobs is defined in a form of SAM projects.

Error recovery is simplified to job resubmission. False starts can simply be re-submitted as new projects.

CDF Production Farm is based on a CAF architecture, Condor batch system is used for job submission and monitoring. This unifies architecture of all the CDF batch computing facilities and allows easy reassignment of the resources if required by the priorities of the experiment. Glide-in technology which currently is in final stage of the beta-testing provides for natural GRID extention of the CDF Production Farm.

The farm control software is modularized and single-threaded to simplify logic of resource management and services at each step from preparation of the input dataset to the final storage.

Each step is independent from the proceeding process which simplifies book-keeping and helps avoiding labor-intensive clean-up procedures.

Having standardized architecture of the Production Farm and implemented SAM-based book-keeping CDF benefited in several ways:

- We have implemented a smaller scale test Farm (so-called "Stage I Farm") which became a place for testing the new farm software, SAM and GRID development. Test activity is happening in parallel with the ongoing data processing on the large Production Farm ("Stage II Farm") but without any interference with it. As both Farms are CAF's, their CPU resources can be dynamically reallocated from one farm to another.

- Should experiment decide to do it, the Production Farm CPU's can be made available for the user analysis jobs running in opportunistic mode. However our default plan is to minimize the default size of the Production CAF such that it provides enough CPU capacity to keep on with the data taking and the calibratioin cycle and reassign part of the analysis CAF resources to the Production CAF when this becomes necessary, for example, in case of data reprocessings.

  This functionality has already been proved working and very useful. In August 2005 CDF successfully reassigned 80 CPU's from the analysis CAF to the Production Farm which allowed to increase the throughput of the Farm up to 20 Mln events per day

- Since the CAF is ubiquitous across CDF, efforts to migrate either general CAF or production processing to the GRID will benefit the other.

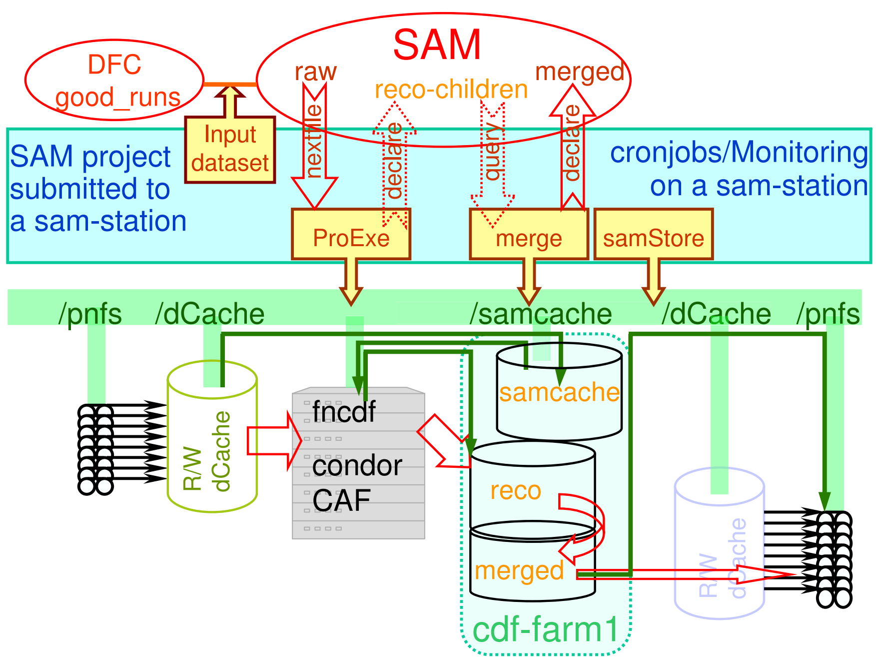

Figure 14: Data flow and job control of a SAM farm. Data are transported by SAM to a file cache accessible to the Condor CAF. Output is sent to a durable storage where concatenation is executed. Merged outputs are declared to SAM and stored to Enstore.

6.1.1 SAM farm architecture

The SAM data handling system is based on SAM and is organized around a set of servers communicating via CORBA to store and retrieve files and associated metadata.

File metadata are stored in the central CDF offline production database (currently fcdfora4.fnal.gov) using the SAM schema.

A task for processing many files is launched as a SAM project. A project is organized

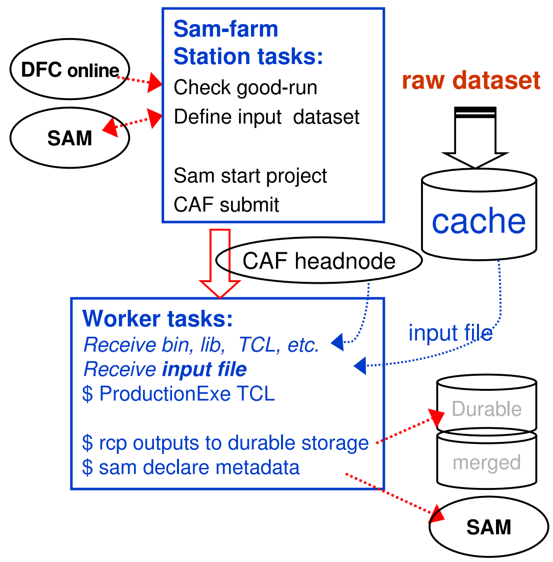

Figure 15: Task flow for a SAM project submitted to a CAF worker. A worker node receives the executable tarball, copies input data file, after processing outputs are copied to durable storage with metadata registered to SAM.

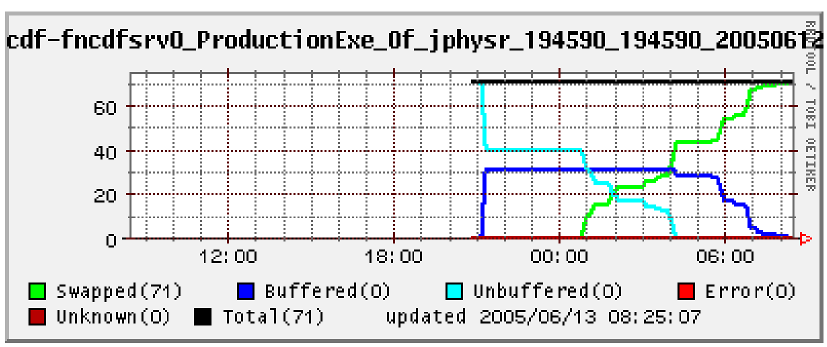

Figure 16: Consumption of files by a SAM project is plotted. Tho total of 71 files in a dataset were requested and quickly "buffered" to CAF workers. The CAF job is configured to use 30 CPU segments. After approximately 4 hours, consumed files are being "swapped". The project is terminated after all files are swapped.

for a user dataset, with a consumer process established to receive data files. File delivery is coordinated such that the events are read only once to all the analysis programs of the project.

Illustrated in Fig. 14 is the hardware architecture and applications for data production with SAM. With the input provided by SAM, disk space is only required for output on durable cache, before concatenation and afterwords to be stored to SAM. The communication with SAM database is conducted by the farm servers configured as SAM stations. The CAF and durable storage are entities easily specified in the job submission, therefore it is flexible to use any facility accessible. To improve bandwidth and file usage, the SAM production farm is configured for direct access to the dCache file system where input files are located. Concatenated output files are transferred directly to Enstore.

Job submission is controlled by applications scheduled on a SAM station. The usage of file metadata is generalized for bookkeeping purpose. The tasks preparing input datasets and data processing in a CAF worker node are illustrated in Fig. 15. The tasks are:

- **Preparation of input datasets :**
  input data to be processed are selected by queries to online DFC records for data quality (good-run) and detector calibration. The input datasets are organized in run sequence of more than 20 files of one or multiple runs for a raw data stream.

- **Start of SAM project, CAF submission :**
  a SAM project is started for a dataset not fully consumed. It is submitted to a CAF. SAM establishes a consumer process to deliver files to CAF workers. From the CAF headnode, workers receive an archived (tar) file containing program binary, library and control TCL cards. Input files are copied to the local scratch area. Files are delivered according to the file consumption status, till all files are delivered. Output of the program are then copied to dedicated durable storage nodes, and the associated metadata are declared to SAM.

The dataset preparation and job submission are all issued periodically by cron jobs.A project monitoring graph on the consumption of data files are plotted in Fig. 16. To prevent exhaustion of computing recourses, permission is required by a monitoring template recording the latest resource status.

SAM farm management is attending processes of datasets. Tracking for individual file is taken care by the SAM consumer process. The operation is therefore reduced to detect incomplete projects and debug. The bookkeeping tasks is reduced from tracking thousands of files in an instance to a few dozens of projects. The monitoring is concentrated on the usage of durable storage, where outputs from CAF are checked and merged in the concatenation process.

6.1.2 Durable storage

Reconstruction jobs running on the Production CAF send their output to to a durable storage implemented as several file servers, which capacity ranges from 2 to 7 TBytes.

A concatenation job is launched upon a threshold of total number of files, these files are then merged (concatenated) into one output file which size is chosen to be close to 1 GByte.

Contents of each input file always goes into one and only one output file. At a price of slightly varying size of the concatenated output this strategy allows to maintain one-to-many relationship between the output and the input files which significantly simplifies the logic of book-keeping and error recovery.

We impose parentage in metadata listing input raw data parents and output children. With a SAM query we find files not yet processed. If a merged output should be reprocessed, we query its parents for preparation of recovery.

The concatenation procedure conducted on the durable storage node is illustrated in Fig. 17. The details are described in the following:

- **Durable cache :** a durable cache is a directory on a concatenation file server where CAF output of the same dataset are stored. In total 41 directories are used for all reconstructed datasets. The files are buffered to a threshold (for example 100 files). A cron job sort them into lists of files in sequence of data taking period. File size of a list is within the desired concatenation range. And the control TCL read by the executable (AC++Dump) is prepared to include these files.

- **Concatenation :** concatenation is running on the file server where the durable storage resides. Output is transported to the "merged" directory ready to be stored to SAM.

- **SAM store :** a cron job checks the total volume of the concatenated files and when it exceeds a certain threshold, for example, 10 GBytes, the files are transferred to the robotic tape storage (Enstore) and their metadata are declared to SAM.

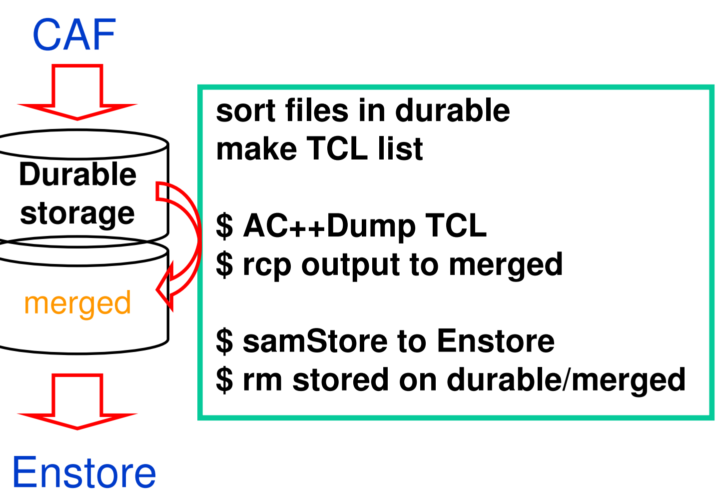

Figure 17: Files in a durable cache are sorted into lists in TCL cards read by the concatenation binary (AC++Dump). The merged files (of size close to 1 GB) are stored to SAM.

As the concatenation process is I/O bound we have chosen to run the corresponding jobs locally on the file server - this reduces the total network load and simplifies the logistics.

6.2 Production Farm operations and Plan

Migration to the new architecture of the Production Farm has been accomplished in sprint'2005 and in the end of May CDF started processing the data taken after teh fall'2004 shutdown.

Reconstruction algorithms have been frozen since winter'2005 and currently they have achieved exceptional level stability. For example, in between Aug 24 and Sep 10 2005 CDF Production Farm has processed about 300 Mln events and we observed 2 crashes of the reconstruction executable in total.

As it is shown in Fig. 18 average event processing time grows up linearly with the luminosity. At average for the spring'2005 instantaneous luminosities of about 0.4e32 (see Fig. 2) event processing time is about 3.8 GHz*sec.

Typically it takes about 3-4 hours to reconstruct 5000-7000 events from one 1GB input file.

The CPU time required for the concatenation is about 3 minutes per GByte on a P3 2.6 GHz file server with the RAID array built from 7200 rpm IDE hard drives and using 3Ware RAID controller.

CDF Production Farm is currently operating with 4 concatenation fileservers. To allow processing of 20 Mln events per day each fileserver needs to provide for the total input-to-output throughput slightly below 10 MBytes/sec, which is equivalent to the total I/O rate of

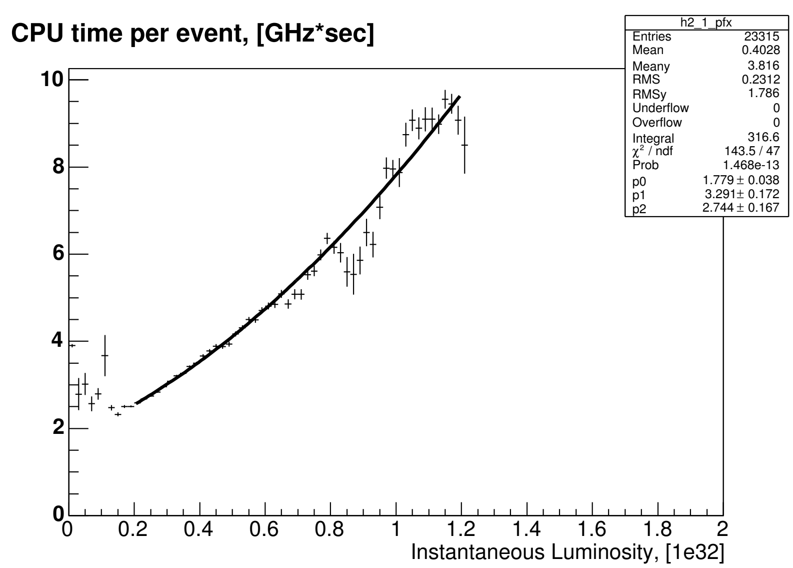

Figure 18: Average event processing time by the CDF offline reconstruction executable (version 6.1.1) as function of the instantaneous luminosity

40 MBytes/sec (20 MB/sec read + 20 MB/sec write).

The network giga-link speed is commonly running at 20 MByte/sec therefore single Enstore mover can transfer to tape more than 1 TByte a day.

In September'2005 CDF Production Farm is averaging about 18 Mln processed events per day, the record number is 21.8 Mln events/day.

Processing 20 Mln events/day on the Production Farm requires approximately 7 STK 9940B tape drives used 100% of time.

Using the model outlined in the previous section, we estimate the total required capacity of the farm as a function of time. The results are shown in Table 2.

Unified infrastructure of the CDF batch computing allows to simplify planning of the budget and the CPU costs needed by the offline reconstruction are also included into the Table 2.

As one can see the projected needs of the CDF offline reconstruction are lower than even rather concervative estimate of the CDF on-site CPU resources. We therefore are planning that the CDF offline reconstruction will always be performed at Fermilab using CPU resources dedicated to the experiment.

7 Databases

CDF utilizes databases for both online and offline applications, and all database servers are running Oracle server software. In the online, the calibrations are both written (by calibration consumers) and used (by Level 3 and other monitoring consumers), and in addition a plethora of other quantities are recorded in the online production database: hardware configurations, run conditions, trigger tables, currents and voltages from the slow controls, etc. In Oracle parlance, the online DB schema are divided into several applications: Trigger, Hardware, Run, Calibration and Slow Control (MCS).

On the other hand, the offline jobs mainly require access to selected calibrations, in addition to rare use of other types of information. The offline jobs, however, require access to data handling information. Thus various data-handling-related schema are unique to the offline databases. CDF is still in the process of migration from DFC[1] to SAM[2] (described in Sec. 5), and as a result both sets of tables are being kept and are being accessed by the CDF analysis jobs. One important difference between DFC and SAM is that the latter requires both read and write access to the offline production instance.

The content of the online production database is replicated to offline production database via the Oracle streams replication.

7.1 Database hardware

CDF currently utilizes Suns for online databases and a combination of Sun and Linux boxes for offline databases. CDF database hardware setup is listed in Table 8. The online production machine, bzora1, is a SunFire V440 with 4 CPUs (each of 1281 MHz). The online development and integration machine, b0dau36, is a Sun Enterprise 450 with 4 CPUs UltraSPARC-II 400 MHz. The online production database is behind the online firewall. It can be accessed freely from the online cluster, however to reach it from elsewhere one needs to be on a machine which is specifically allowed to connect. (An example of such machine is fcdflnx2.)

The offline production machine is fcdfora4 which is a Sun V880 with 8 900 MHz processors, 32 GB RAM with about 1 TB of fiber channel disk drives, and Gigabit Ethernet. fcdfora1, an older Sun Enterprise E4500, hosts the offline development and integration Oracle instances. Since April 2005 the content of online production database (except for the slow controls) and Data File Catalog are replicated to more powerful machine, fcdfora6 DellT PowerEdgeT 6650 System with 8 Intel(R) XEON(TM) MP CPU 2.00GHz

The access to the offline production database is not restricted. However, the DB API used by the CDF offline jobs ‘throttles’ new connections to any of the offline Oracle instances by counting the number of active and total connections of both the user who is trying to connect and all other users – and then refuses to open a session in Oracle if any of the limits have been exceeded. In FY2004, and especially in FY2005, an increasing number of user Monte

[1] Data File Catalog
[2] Sequential Access through Metadata

name	OS	CPU	RAM	Disk	Oracle
bzora1	Solaris 2.9	4×1281 MHz USparc	16 GB	1.2 TB	9.2.0.6.0
b0dau36	Solaris 2.8	4×400 MHz USparc	4 GB	1.2 TB	9.2.0.6.0
fcdfora4	Solaris 2.8	8×900 MHz USparc	32 GB	1.3 TB	9.2.0.6.0
fcdfora1	Solaris 2.8	2×400 MHz USparc	1.25 GB	540 GB	9.2.0.6.0
fcdfora6	RH AS 3	8×2 GHz Xeon	16 GB	2 TB	9.2.0.6.0

Table 8: Database hardware and software configuration

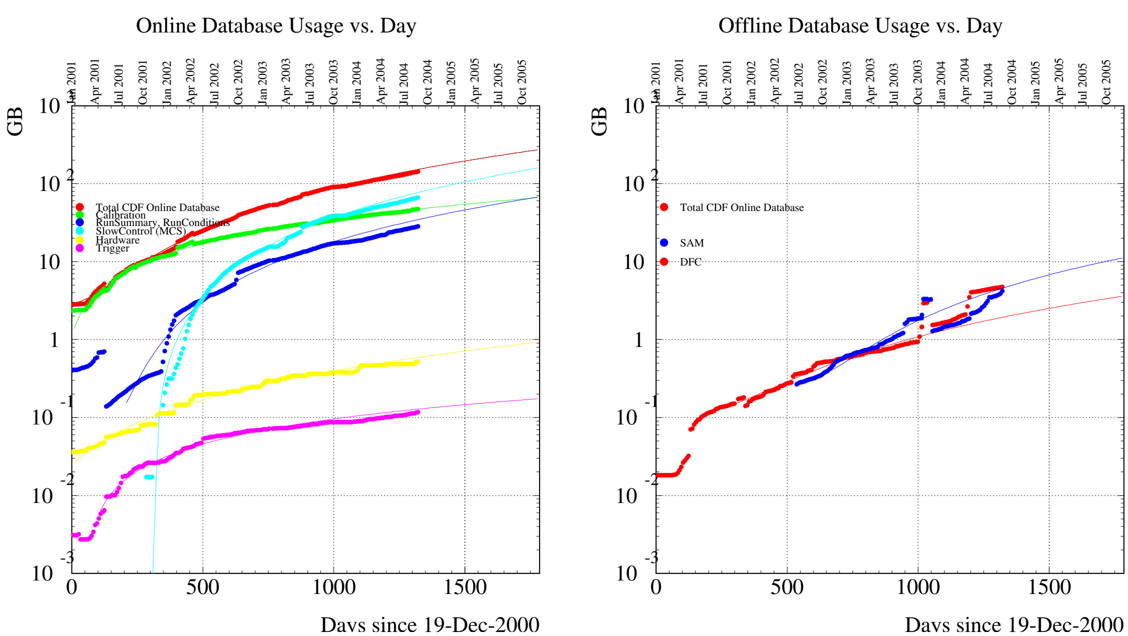

Figure 19: CDF DB space usage on online and offline production Oracle instances. Plot for offline instance shows only read/write applications: SAM and DFC.

Carlo simulation jobs ran against this limit, which caused the jobs to fail. The DB group is addressing this problem by deploying the Frontier DB interface (described in Section 7.4 below), which dramatically reduce the need of CDF offline jobs to connect to Oracle.

7.2 Table Space Usage

The amount of data used by existing application is constantly monitored. Disk space usage as the function of time is used to make projections of space needed for application in the future. Example plots for online and offline production databases are shown in Figure 19.

7.3 Propagation of Database Content

The online data logger and production farms need continuous access to the offline production database in order to log and reconstruct raw data. Increased analysis activity accompanied by substantial growth of CPU power led to several incidents when database and system resources could not handle the demand. This issue was addressed by developing a strategy of distributing the database content via replication on site. The replica copies of the database are read-only instances accessed by the majority of users; the production farms and online data logger thus have an exclusive access to the primary offline production database. As pure read-only databases the backup costs are minimal. However, for the clients of the replica, there is a fail-over to the offline production instance in case of emergency or maintenance work on the replica or the machine which hosts it.

The first replica, cdfrep01, hosted by fcdflnx1, was in use from the summer 2002. The replication from proceeds via Oracle read-only replication from the originating either online or the offline production databases depending on the application.

In April 2005 the CDF DB and CD/DSG groups replaced the old online production machine with bzora1, and the offline replica machine fcdflnx1 by a newer and much more powerful fcdfora6. At the same time, the replication was switched to the Oracle streams replication. Oracle streams allow the data propagation to proceed in sequential mode, thus avoiding firewall issue, and, even more importantly, reducing the load on the source database machine. Streams also allows automatic propagation of DDL changes to replica sites. Online Database are replicated to offline. Then from offline all on-line data and SAM and DFC, BOOKS are replicated to another CDF replica. Offline and Replica share the common read service. If CDF replica is down then there is automatic failover to CDF offline database.

7.4 Support of computing at remote sites

The CDF's database group and the DBS group from Fermilab's CD have developed an N-tier database access patterned after the DAN used by D0. Although the N-tier access does not resolve all Oracle licensing issues, it provides a local caching (or "secondary sourcing") of data from offline production database resulting in more efficient use of computing resources at the remote sites.

The Frontier DB is a new system for the distribution of frozen database content which utilizes standard Web tools connected into a multi-tier topology. At the moment, it is CDF's best candidate for providing for remote database access from DCAFs.

An example of how Frontier DB handles a request for a typical calibration table is shown in Fig. 20. The gain comes from the use of the Squid server to cache the response of the Frontier servlet. The fact that the client has been code-generated ensures that all requests for a specific calibration table produce one and the same HTTP string, and thus only the first such request actually reaches the Tomcat and causes both Tomcat and Oracle to perform work. All subsequent requests are handled solely by the Squid, which simply delivers a cached already prepared response.

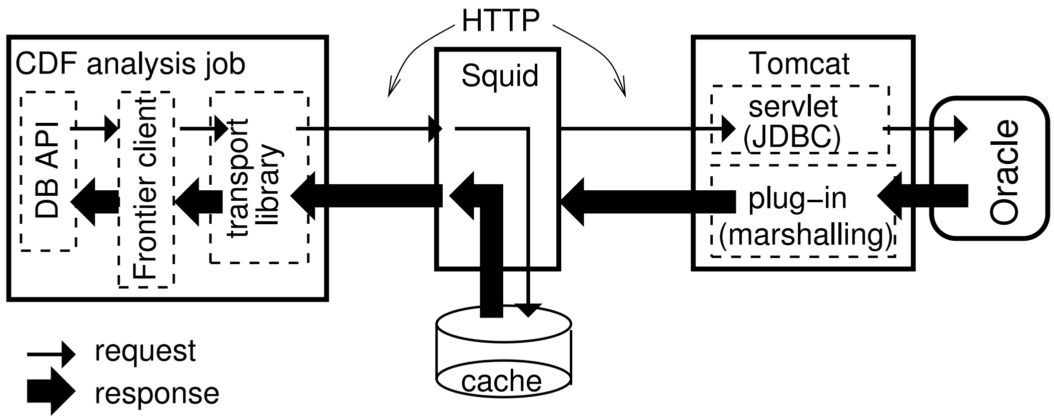

Figure 20: A sketch of the execution of a Frontier DB request. When the DB API discovers that it needs a certain calibration table, it calls the Frontier client. Just like an Oracle (OTL) client, the Frontier client is code-generated. The Frontier client specifies the order and the types of the fields, but it delegates the details of the data decoding to the Frontier transport library, which uses libCURL to send a HTTP request and retrieve the response. The request passes through a Squid server, and reaches the Tomcat server which runs a Frontier servlet. The Frontier servlet uses JDBC to query Oracle (offline replica DB). Oracle's response is passed onto a plug-in specific to this table, which also has been code-generated along with the Frontier client and which calls subroutines from the Frontier transport library (frontier_client.so) to ensure data consistency between the servlet and the client. The response of the Frontier servlet is cached in the Squid, so that every other request for this calibration table will retrieve the cached response.

The essence of the proposed implementation is the wide-spread deployment of Squid servers, preferably as close to clusters of worker nodes as possible. We assume that every remote DCAF will have at least one local Squid, and that many university groups will elect to have local Squids as well. At Fermilab, there is a system of four former CAF worker nodes which have been reconfigured to run both a lowest-tier Frontier server (in Tomcat) and a Squid cache. All four Tomcats can connect to the offline replica database (and can fail-over to offline production just like any other oracle client), and the four Squids have been set up in a load-balancing way. These four machines with Tomcats and Squids constitute the so-called Launchpad, which is the main Frontier access point for both on-site CDF jobs and off-site Squids. True to its name, the system has an N-tiered topology: the remote Squids connect to one of the four Squids with Tomcats in order to utilize an already cached data.

7.5 DB Activities for the FY2006

Here is the list of upcoming activities of the Database group. The list is not exhaustive, however, it indicates the priorities in the next year.

- The DB group will make an effort to transition to Oracle 10g in the near future, and preferably complete both the online and the offline during the upcoming fall shutdown 2005. This is the group's highest priority.
- Closely watch fcdfora4 load due to SAM schema queries by individual users.
- Help CDF users while the Frontier is moving into production.
- Propagating tnsnames.ora was a nightmare. The lessons learned need to be documented. (In particular, the CDF should clean up the scrips which package the tnsnames.ora into the tar files to be ran on the Grid.)
- The DB group will not make any effort to split the slow controls onto a separate instance, as the online group does not seem to think that this would benefit CDF operations during the recovery from a catastrophic failure of bzora1, whereas it would require a lot of effort on their part to adjust the slow controls software.
- ODBC back-end should move into production.
- Support of freeware databases (MySQL, Postgres) is not a high priority for CDF since the Frontier will supersede them. (Frontier is leaner, faster for read-only data, and does not require any administrative effort at the remote sites, and is thus more attractive for DCAFs.) However, the CDF operations occasionally involve some of the freeware databases, so a need for a low-level of support cannot be ruled out in the future.
- Much of the activity on fcdfora4.fnal.gov database instance cdfofpr2 does not use the AC++ database monitoring code, and so information about database instance cdfofpr2 is limited.
- CDF needs to increase the involvement of physicists in the DB operations, especially having to do with the use of the DB API from the user analysis and Monte Carlo simulation jobs. In particular, a CDF physicist need to examine the DB access from TrigSim++.

7.6 DB budget

The existing load from users' jobs running on Fermilab CAFs is well handled by the replica machines. The offline production machine serves exclusively the production farm and is loaded lightly. Therefore with exiting DB setup supplemented with load balancing between offline production and cdfstrm1 we should manage to handle ever increasing CDF load during the lifetime of the experiment. The Frontier solution which is entering production this fall will allow us to shift load from expensive machines running Oracle servers to commodity Linux boxes running Frontier components. Approximate breakdown of database spendings are given in Table 9. Starting 2006, after the hardware for Frontier system is bought, we foresee only maintenance costs.

FY	DB CPU (n-ways)	DB Disk (TB)	Cost ($M)
03			0.15
04	2	4	0.07
05	6	1	0.05
06	2	2	0.03
07	2	2	0.03

Table 9: Database CPU and disk procurement plan. The fiscal year, the number of n-way Linux boxes purchased that year for DB machines, the TB of disk purchased and the cost.

8 Networking

8.1 CDF Networking

In 2005 in response to power and cooling limitations in the Feynman Computing Center (FCC) CDF has moved a large number of CAF worker nodes to another location - GCC.

Currently most of the worker nodes are located in GCC with the remaining nodes also planned to be moved over there.

FCC building still hosts all the disk servers, this separation of the worker nodes and the data servers requires more careful assessment of network topology.

The heart of the CDF offline computing network is the CAS switch, a Cisco 6509, located in FCC2.

- It has 4 10 GBit connections, one port is reserved for the site up-link and the other three are connected to CAF switches: one located in FCC1, one in FCC2, and one in New Muon.
- Fcdfsgi2 is currently connected to this switch via 5 Gbit connections, 1 for interactive use and 4 for Enstore.
- The CDFEN Enstore robot currently has 18 GigE connections to the offline switch for the movers for T9940B drives.
- The stage 1 CAF file servers use 15 GBit connections and the CAF stage 1 worker nodes use 67 FE connections.

In 2005 the CDF network has been upgraded, its schematic diagram is shown in Figure 21.

In order to support the new CAF work nodes and the new farm nodes planned for GCC CDF needs to purchase a new Cisco 6509 switch with 160 copper ports and 4 10 GBit fiber ports for up-link.

For each of FY06, FY07, and FY08 we plan on purchasing additional CAF worker nodes and file servers with the necessary network connections. The proposed 6509 for HDCF has sufficient capacity for estimated FY06 CAF worker nodes acquisitions. Additional worker node purchases in FY07 and 08 will require an additional switch. The switches in FCC have sufficient capacity for disk server purchases in FY06 and 07.

In table 10 we estimate the cost of this networking by assuming a Moore's law like decrease: networking costs that drop by a factor of 2 every 18 months. In practice networking costs have dropped much more slowly than Moore's law. One issue that was not foreseen in previous computing plans was the need to duplicate network infrastructure in satellite buildings.

8.2 Trailer LAN

The networking in the trailers has not been upgraded in many years, and the networking group has recommended an upgrade for each of the past 4 years. The available network

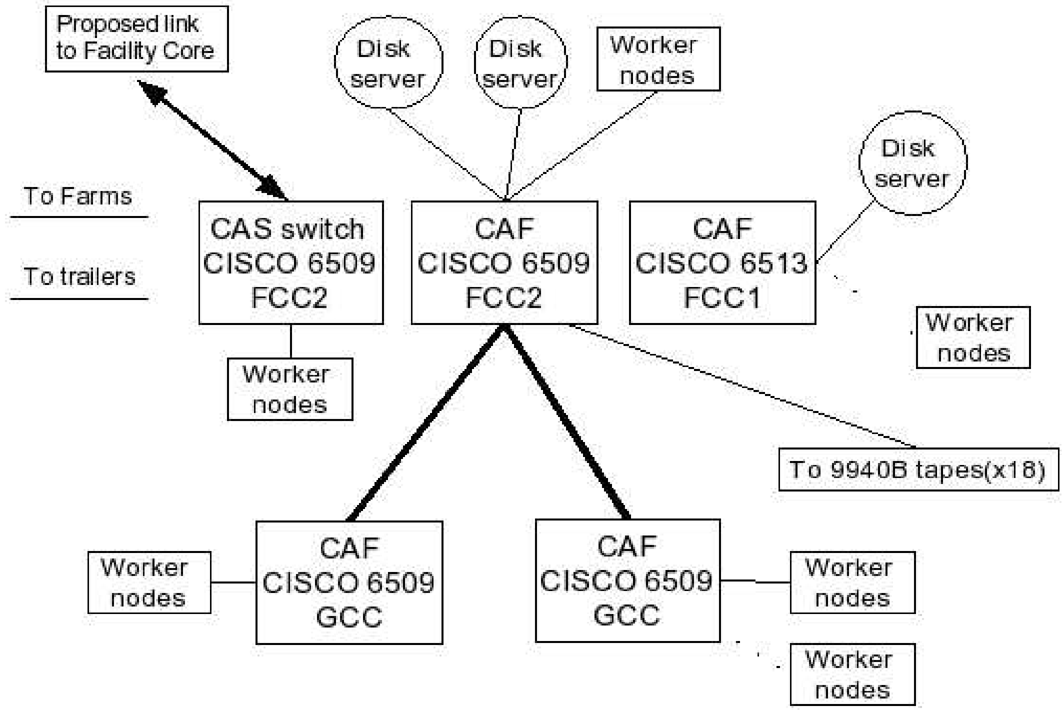

Figure 21: A schematic view of the CDF network architecture in 2005.

resources have consistently been used for CAF and computing center needs. The CDF trailers LAN currently supports 100 Mb/s connections to the majority of CDF offices, and this lags behind the current network capabilities of desktop Ethernet cards which are 1 Gb/s, and restricts the data transfer rates for existing file servers in the trailers. Currently multiple satellite switches are used to extend the ports available on the trailers 6509 switch, in an architecture that lowers the bandwidth capacity of many offices. The infrastructure in the trailers is currently primarily fiber. The new office building is wired with copper.

All the CDF 65 series switches have been upgraded to the newest supervisory module except the 6513 used in the trailers. This switch currently has two trunked gigabit links to the CAS switch, which will be oversubscribed. This switch also serves as the up-link from the new CDF office building switches to the CAS switch in FCC. In order to provide a 10Gb up-link from the trailer switch an upgrade is needed. This year CDF should provide a limited number of gigabit ports and upgrade the switch. This will allow the gigabit infrastructure in the offices to grow in the future. We also plan to upgrade the supervisor module to Sup720.

With the copper infrastructure in the new CDF building, providing gigabit is somewhat easier. For a small initial investment with the possibility of upgrade in the future, a Cisco 4506 seems like an appropriate choice.

FY	FCC Cost ($M)	Trailer Cost ($M)	Total Cost ($M)
05	0.18	0.07	0.25
06	0.12	0.09	0.21
07	0.04	0.04	0.08

Table 10: LAN procurement plan. The fiscal year, cost of Fermilab computing center networking, cost of CDF trailers networking and total cost.

8.3 WAN

In FY03 the OC3 connection between Fermilab and ESNET was upgraded to OC12 with a capacity of 622 Mb/s. In 2004 FNAL purchased a fiber connection to the StarLight hub in Chicago. This has provided 2 1 GBit and 1 10 GBit research networks to the lab. While the main traffic for the site will continue to go through the ESNET connection, research projects and schedule-able data transfers can use the higher performance fiber connection. The port on the CAS switch 10 GBit blade reserved for the up-link should be installed with a 10 GBit GBIC.

8.4 Proposed Budget For 2004

The proposed purchases for this year along with their estimated costs are given in Table 11.

Description	Quantity	Cost	Total Cost
Cisco 6509 switch	1	\$115K	
Starlight link upgrade to 10 GB	1	\$5K	
Cisco 7600 Supervisor 720 Module	1	\$23K	
Expanded 1000B-T support		\$20K	
FCC link upgrade to 10GB		\$17K	
Cisco 4506 switch	1	\$32K	
Total		\$212K	

Table 11: Proposed network procurements for 2004.

This does not include the cost of small fiber to copper switches in the trailers, but total cost for enough switches to fully utilize the 16 available gigabit ports should be about \$10k.

If any networking budget is available, CDF should purchase a 48 port gigabit module for the upgraded trailer 6513 at a cost of \$9,100. This will allow all the offices currently connected with copper ports to upgrade to gigabit this year. Otherwise, the module will be included in the FY07 budget.

8.5 Proposed Networking Plans for 2006

After the additional switch procurement in 2005 CDF has sufficient networking capacity for a 2006 hardware procurement similar in size to the 2005 procurement.

The network budget for new infrastructure in 2006 will mainly be applied to network blades for GCC, where an additional switch needed; upgraded gigabit infrastructure for offices, both the trailers and the new office building, which more gigabit fiber modules can be used from the CAS and CAF switches and additional copper gigabit switches are needed; and upgrades of the modules in the CAS and CAF switches in FCC to replace low density blades with higher density copper gigabit modules.

The item that needs to be watched in 2005 is the over-subscription on the 2 10Gb links between FCC and HDCF. The plan of the computing division to to locate equipment that requires uninterruptible power, like disk servers, in FCC and high power density equipment, like worker nodes, in HDCF.

Currently 2 10Gb links are proposed. As CDF moves from 30-40TB per day of data served to analysis applications to 80TB of data served per day, the 10 gigabit links will begin to see high utilization. It is possible to add additional 10Gb links between HDCF and FCC by taking advantage of open 10Gb ports on the CAS and CAF switches and adding multiple routes between an HDCF CAF switch and FCC. This will require a reorganization of the subnet used in the CAF and may require the acquisition of routing modules for the CAF switches that host disk servers. The utilization of the current links should be monitored and an upgrade should be reserved as an option.

In 2006 there will be additional CAF acquisitions for both HDCF and FCC. In HDCF there will not be any network ports available and another 6509 (or 2006 equivalent) will be required. The networking capacity in FCC should be sufficient, provided some of the current worker nodes hosted there are retired. By 2006 the separation of disk servers and worker nodes between FCC and HDCF will be complete and if the CDF networking between the buildings has not been upgraded to multiple 10 GBit links it will probably need to be.

Moving towards the GRID-based analyses will require 10GBit upgrade of the Starlight link.

9 Offsite Computing

9.1 Status and Perspective

Offsite computing is now an important reality of the CDF computing environment; offsite resources now account for approximately 50% of all the available CPU resources. The motivation and history of the plan that led to this situation are described in the 2003 and 2004 versions of this document and will not be repeated.

However, since 2003, the HEP world has become much more Grid-oriented. Common computing pools, accessible via Grid tools (mostly Globus based), now represent the majority of HEP resources around the world. We therfore decided to stir the CDF computing infrastructure much more in that direction than it was originally envisioned. CDF can gain by using these common pools in two ways:

1. Several institutions have expressed the intention of limiting the amount of new CPU resources for CDF, unless we put them in common pools. So if we want to grow, and we need to, we have to shift away from the dedicated pools we were using in the past.

2. There are lots of unused CPU cycles in the current HEP common pools, and the situation is expected to stay this way at least until the startup of LHC. Most of these pools are owned and operated by LHC experiments and are used for data challenges and other scalability tests that need a full scale system to be useful. Out of those periods, a significant fraction of the resources stay idle, and CDF could well use them for its advantage.

In practical terms, we expect CDF to expand from the current proprietary clusters to incorporating shared resources in a step-by-step process:

1. Enable CDF institutions with shared pools to access those resources, both the CDF share and any other idle resource.

2. Exploit idle CPU resources on larger Grid sites, where CDF has no institutional presence.

3. Exploit idle CPU resources on also the smaller Grid sites, possibly grouping them together in a single virtual site.

4. Deploy a Grid-wide portal for submission of MC jobs.

5. Optimize the use of opportunistic Grid CPU resources by dynamically allocating and using spare disk resources.

6. Deploy a Grid-wide portal for user submission, that will do intelligent site selections with CPU-data collocation, too.

All the above must be achieved without the need for users to change their habits. Minor correction to their scripts are considered acceptable, major rewrites are not. The current monitoring environment is also considered very important, and very little functionality, if any, should be lost in the process.

In FY-05 we focused on point 1, i.e. we have a working version of a CAF portal, called GlideCAF, that can exploit any local shared pool. Its name comes from the Condor glide-in mechanism that allows us to create a virtually private cluster (VPC) that grows and retracts by the resource sharing rules of the common pool. The VPC is instead completely owned and managed by us, so we are free to manage our CDF users the way it best suits the collaboration, without any interference and/or coordination with the common pool administrators. The VPC also looks and feels like a regular Condor pool, so all the features of the Condor-CAF are preserved.

For FY-06 we are looking forward to implement the other steps listed above. The solutions needed to make steps 2-4 are clear in our mind and are based on advanced features of Condor. We are also teaming with CMS, which is following a similar path, to speed up the development cycle. Additionally, we have a parallel project going on in Europe that is trying to use the gLite Resource Broker; although still in development, it holds great promise, but will probably be limited to the LCG/EGEE Grid only.

Steps 5 and 6 are a little more complex, and we are still exploring various options. CDF has standardized on SAM for its data handling, so this helps us somehow in the task. We will explore all the available possibilities and will choose the one that offers us the most resources with the least amount of development and maintenance effort.

More details about the planning for the CDF offsite computing can be found in the CDF Note 7817.

In the transition to shared resources, we must not forget the currently deployed offsite proprietary clusters that are still offering a large share of the CDF resources. Some of them are going to be converted to shared clusters in the short term, while other may decide to remain in their current incarnation for quite some time. From the CDF user point of view they look and feel exactly the same, so we are committed to support both of them.

9.2 Status of Offsite Resources as of Summer 2005

While some CDF collaborators own CDF-reserved computers, other share access to largish facilities with other experiments. This makes it almost impossible to tell a-priori how much CPU power CDF physicists can use offsite, although we do have a guaranteed minimum on each and every site. Additionally, several institutions give additional priority to their own members, so there is not an a-priori knowledge as to how much usage generic CDF users will obtain, making the picture even more complex. However, a-posteriori it is easy to find out how much each and every site has given both to the CDF collaboration as a whole and how much has been given to the generic CDF user.

In the following table, we present a snapshot of offsite hardware resources available for CDF by summer 2005. In this table, we only listed institutions that allow access to all CDF members. The German GRIDKA site is an exception because while being the institution policy it is still technically problematic since access has to be done via LCG GRID software. The situation is expected to change before the end of 2005, with the introduction of a GlideCAF portal.

Institution	Total CPU (GHz)	Min CPU (GHz)	Local CPU (GHz)	Disk (TB)
Canada (TorCAF)	560	50	0	10
Canada (MC farm)	1080	240	0	MC farm
Germany (GRIDKA)	3000	220	0	30
Italy (CNAFCAF)	1200	600	400	40
Japan (Tsukuba)	340	340	190	10
Korea (KorCAF)	175	175	0	0.6
Spain (CANCAF)	50	50	40	1.5
Taiwan (ASCAF)	150	150	0	3
Rutgers (RUTCAF)	60	60	0	12
MIT (MITCAF)	300	300	0	4
UCSD (SDSCCAF)	450	260	0	5
TOTAL	7000	2500	650	120

Table 12: Computing resources available offsite to CDF users by summer 2005.

In addition to what is shown in the table, CDF groups in Spain at Barcelona, in France at Lyon, and in the USA at Wisconsin are setting up GlideCAF portals to the local Grid pools to use opportunistically any CPU not used by other virtual organizations. These sites are expected to be up and running well before the end of 2005.

FY-05 has also seen an increase of disk resources available offsite. While this increase was very welcome, much more is needed to make the offsite data analysis a viable solution. For comparison, the Fermilab CAFs have approximately 100Gb of disk per GHz of CPU, while the average for offsite sites is as low as 20Gb per GHz.

Most offsite institutions also have a high speed local connection to the Internet, so access to those facility can be highly efficient. However, experience shows that effective throughput on WAN can be limited by many hard-to-find bottlenecks. Serving data on demand is a possible, but still not a very efficient solution, so pre-staging of data will be the way to go for most of the data intensive tasks, at least for the next year.

9.3 Offsite MC Production

At present, most of the CDF MC generation is performed offsite, both the organized MC production and the user MC production. Some users are still generating their events on the Fermilab CAFs, but those are just small personal productions for users that don't need much resources. Anyhow, we are monitoring the situation and telling such users to move offsite.

The move from dedicated to shared pools of resources should be completely transparent for the users as long as the CDF software distribution is accessible from all the nodes. At major CDF managed sites we expect this to always be the case, and it certainly is the case for all the shared pools we set up in FY-05.

However, the opportunistic use of Grid resources a much more tricky business; most of the sites will not have direct access to the CDF software distribution. To use these resources at best, we are once again using a step-by-step approach:

1. We have installed the CDF software distribution in a couple AFS cells, so we can run on any site that has AFS clients deployed. We will also try to dynamically install the CDF software distribution on any major site that will allow us to do so.

2. Organized MC production can create self contained tar balls that do not rely on the CDF software distribution. This requires a couple of days of work every time something major is changed, but is considered worthwhile for the additional resources this can gain us.

3. We are looking at ways to have the CDF software distribution accessible from any network attached node in the world. We have found some promising technology and we are now trying it out.

9.4 Offsite Data Analysis

While most of the data analysis is still being performed at the Fermilab CAFs, a significant fraction has moved offsite. Several offsite sites now have an established procedure to import large data sets to local SAM stations, making them available to CDF users for analysis, so relieving CPU load and data access congestion from FNAL's CAF. We have experienced that it is more effective to preload specific data sets, lock them on local cache disk, and advertise their availability to users, rather then import data on demand according to random analysis jobs and end with a lot of cache misses.

The shift from dedicated to shared CPU resources has been painless on established CDF sites; jobs are being executed on shared CPU resources, but the disk space is still owned and managed by CDF. We expect this trend to continue on newly deployed sites with strong CDF involvement.

Opportunistic exploitation of Grid resources for data analysis is much more problematic. In addition to the possible lack of the CDF software distribution, described in the previous section, we are now facing also the problem of efficient data access.

Our current roadmap to opportunistic Grid data analysis is as follows:

1. No data analysis on opportunistic resources, at first. We will use those resources for MC generation only, thus offloading the sites with large data pools.

2. On sites that have good network connectivity to an established SAM station, we will try to access data from that SAM station. If everything goes well, we will install regional SAM stations to cover most of the Grid sites we have access to.

3. The final goal is to use disk resources that Grid sites give us on opportunistic base. We have no obvious solution for this at the moment, but we do want to explore the option.

9.5 The Financial Side

The CDF International Finance Committee has been debating at length the formalization of foreign contribution to CDF costs. The current position of the Committee is that the CDF's plan to have 50% of analysis work done offsite is reasonable and it is a matter of fact that CDF is already very near that level, with several countries having contributed the necessary resources to reach that goal. On the other hand, such contributions has been on a voluntary basis, in a best effort spirit, and it is quite likely that it will stay that way and there will be no MOU-kind document. At the same time proper accounting of the usage of remote resources is perceived to be fundamental for a fruitful collaboration both as guideline for efficient usage and acknowledgment of the contribution.

Up to today, even in lack of such a formalization, CDF has nevertheless received substantial financial contribution by several foreign countries, the biggest being Italy and Canada. Several US universities have also contributed to the common goal, either by contributing CPU and disk resources, like in the case of MIT and Rutgers, or by hosting common resources, like in the case of UCSD.

It is CDF's first priority to preserve all positive sides of how things have been working till now, and therefore to be very cautious and careful in defining a brand new policy.

References

[1] Dave McGinnis,
FERMILAB TEVATRON OPERATIONAL STATUS, talk at PAC05 conference,
http://beamdocs.fnal.gov/cgi-bin/public/DocDB/ShowDocument?docid=1839

[2]